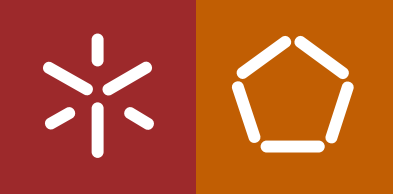




Toni Daniel Neto Machado

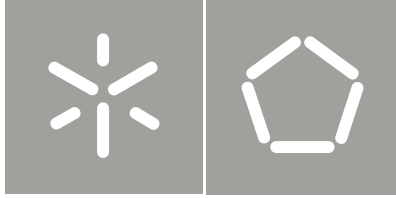
**Transporte de objectos por equipas de robôs móveis autónomos:** implementação e validação de uma arquitectura de controlo distribuída, baseada em sistemas dinâmicos não lineares.

**Universidade do Minho**  
Escola de Engenharia









**Universidade do Minho**

Escola de Engenharia

Toni Daniel Neto Machado

**Transporte de objectos por equipas de robôs móveis autónomos:** implementação e validação de uma arquitectura de controlo distribuída, baseada em sistemas dinâmicos não lineares.

Dissertação de Mestrado  
Mestrado em Electrónica Industrial  
Área de Automação e Robótica

Trabalho efectuado sob a orientação da  
**Doutora Estela Guerreiro Silva Bicho Erlhagen**

## DECLARAÇÃO

**Nome:** Toni Daniel Neto Machado

**Endereço electrónico:** tmachado@dei.uminho.pt

**Telefone:** 917901823

**Número do Bilhete de Identidade:** 12140810

**Título da dissertação:** Transporte de objectos por equipas de robôs móveis autónomos: implementação e validação de arquitecturas de controlo distribuídas, baseadas em sistemas dinâmicos não lineares

**Orientadora:** Doutora Estela Guerreiro Silva Bicho Erlhagen

**Ano de conclusão:** 2007

**Designação do Mestrado:** Mestrado em Electrónica Industrial – Automação e Robótica

É AUTORIZADA A REPRODUÇÃO INTEGRAL DESTA DISSERTAÇÃO APENAS PARA EFEITOS DE INVESTIGAÇÃO, MEDIANTE DECLARAÇÃO ESCRITA DO INTERESSADO, QUE A TAL SE COMPROMETE.

Universidade do Minho, \_\_\_\_ / \_\_\_\_ / \_\_\_\_

**Assinatura:** \_\_\_\_\_

# Agradecimentos

A realização desta dissertação nunca teria sido possível sem uma conjugação de esforços, sendo nesta fase importante referenciar uma entidade e algumas pessoas.

Esta dissertação foi suportada, em parte, através de Bolsas de Investigação Científica no âmbito do projecto: “*CoopDyn: Synthesis of Cooperative Behavior in Multi-robot Systems: a Nonlinear Attractor Dynamics Approach*” (ref.: *POSI/SRI/38051/2001*) financiado pela Fundação para a Ciência e a Tecnologia (FCT) e pelo Fundo Europeu de Desenvolvimento Regional (FEDER).

Devo agradecer com uma atenção especial à minha orientadora, doutora Estela Bicho, pelo desafio que me lançou e por todo o apoio prestado.

Seguidamente agradeço ao Rui Soares, pois a arquitectura de controlo utilizada nesta dissertação foi desenvolvida pelo mesmo em ambiente de simulação no âmbito do seu doutoramento.

Agradeço ainda aos meus colegas e amigos de laboratório pelo bom ambiente e ajuda no trabalho, sendo eles a Eliana Silva, o André Moreira, o Nzoji Hipólito, o Rui Silva, o Manuel Carvalheira, o Sérgio Monteiro e o Luís Louro. A este último, o meu muito obrigado pelo seu apoio aquando da implementação das arquitecturas de controlo.

Agradeço também todo o apoio prestado pelos técnicos da oficina do departamento de Electrónica Industrial, o Carlos Torres, o Joel Almeida e a Ângela Macedo.

Por último, mas não menos importante, devo agradecer à minha família e à minha namorada todo o apoio nos bons e maus momentos passados, e também pela paciência que tiveram em alguns momentos menos compreensíveis da minha parte.

# Resumo

Esta dissertação tem como objectivo a implementação e validação de uma arquitectura de controlo, baseada em sistemas dinâmicos não lineares, para uma equipa de robôs móveis autónomos que em conjunto devem transportar um objecto de grandes dimensões desde uma posição inicial até uma posição de destino, e em simultâneo evitar colisões com obstáculos estáticos e/ou dinâmicos que possam encontrar no seu caminho. A arquitectura de controlo foi desenvolvida por um elemento do Grupo de Robótica Autónoma e Sistemas Dinâmicos, do departamento de Electrónica Industrial, da Universidade do Minho, no âmbito do seu doutoramento.

Os robôs não têm qualquer conhecimento prévio do ambiente que os rodeia. A arquitectura de controlo é estruturada em termos de comportamentos elementares. Os comportamentos individuais e a sua integração são gerados por sistemas dinâmicos não lineares.

Para a implementação da arquitectura de controlo foram utilizadas equipas de dois e de três robôs móveis autónomos, tendo sido os resultados documentados em vídeos. Estes resultados mostram o bom desempenho dos robôs em vários cenários, quer com obstáculos estáticos, quer com dinâmicos. O comportamento resultante é estável e as trajectórias são suaves, isto porque à medida que a informação sensorial e/ou comunicada varia, os sistemas dinâmicos adaptam-se, gerando soluções atractoras adequadas.



# Abstract

The goal of this dissertation is the validation of a distributed control architecture, based on non-linear dynamical systems, for a team of autonomous mobile robots that jointly carry a large object from a starting position to a position of destination, and simultaneously avoid collisions with obstacles static and/or dynamic that can encounter in their path. The architecture was developed by an element from the Dynamical Systems and Autonomous Robotics Group of the Department of Industrial Electronics, from University of Minho, this work was carried out for his *PhD*.

The robots have no prior knowledge whatsoever of the surrounding environment. The control architecture of each one of the robots is structured in terms of elemental behaviours. The individual behaviours and their integration are generated by non-linear dynamical systems.

The implementation of the control architecture was performed on teams of two and three robots, the results were documented in video. These results show the good performance of robots in several environments, with static and/or dynamic ones. The resulting behaviour is stable and the trajectories are smooth, this is achieved because the dynamical systems adjust to changes in the information provided by the sensors and/or the information communicated between the robots, generating adequate attractor solutions.





# Conteúdo

<b>1</b>	<b>Introdução</b>	<b>1</b>
1.1	Transporte de objectos por equipas de robôs . . . . .	3
1.1.1	Soluções para sistemas duo-robô . . . . .	3
1.1.2	Soluções para sistemas multi-robô . . . . .	8
1.2	Objectivos da dissertação . . . . .	13
1.3	Estrutura da dissertação . . . . .	14
<b>2</b>	<b>Princípios básicos para a geração de comportamentos usando sistemas dinâmicos não lineares</b>	<b>15</b>
2.1	Variáveis comportamentais . . . . .	16
2.2	Dinâmica comportamental . . . . .	17
2.3	Pontos de equilíbrio . . . . .	18
2.4	Força dos pontos de equilíbrio . . . . .	21

<b>3</b>	<b>Robôs móveis</b>	<b>23</b>
3.1	Os robôs e as suas tarefas . . . . .	23
3.2	Características . . . . .	25
3.3	Comunicação . . . . .	26
3.4	Cinemática . . . . .	27
<b>4</b>	<b>Projecto do suporte</b>	<b>29</b>
4.1	Antigo suporte para o objecto . . . . .	29
4.2	Suporte para o objecto . . . . .	30
4.2.1	Juntas prismáticas . . . . .	32
4.2.1.1	Leitura de uma junta . . . . .	32
	Programação do PIC 16F874 . . . . .	34
4.2.1.2	Conversão série-paralelo da leitura da junta . . . . .	34
	Programação do PIC 16F876 . . . . .	36
4.2.2	Junta rotacional . . . . .	36
<b>5</b>	<b>Arquitectura de controlo do robô <i>Leader</i></b>	<b>39</b>
5.1	Dinâmica de atractores para a direcção de navegação . . . . .	39
5.1.1	Comportamento <i>dirigir-se para o alvo</i> . . . . .	40
5.1.2	Comportamento <i>evitar colisões com obstáculos</i> . . . . .	42

5.1.3	Integração dos dois comportamentos, <i>dirigir-se para o alvo e evitar colisões com obstáculos</i> . . . . .	45
5.2	Controlo da velocidade . . . . .	46
5.3	Hierarquia das taxas de relaxação . . . . .	48
<b>6</b>	<b>Arquitectura de controlo para o(s) robô(s) <i>Helper(s)</i></b>	<b>49</b>
6.1	Dinâmica de atractores para a direcção de navegação . . . . .	50
6.1.1	Para a equipa de dois robôs . . . . .	51
6.1.1.1	Comportamento <i>Forward/Backward</i> . . . . .	53
6.1.1.2	Comportamento <i>Turn</i> . . . . .	56
6.1.1.3	Comportamento <i>Column</i> . . . . .	58
6.1.2	Para a equipa de três robôs . . . . .	59
6.1.2.1	Comportamento <i>Forward/Backward</i> . . . . .	61
6.1.2.2	Comportamento <i>Turn</i> . . . . .	65
6.1.2.3	Comportamento <i>Column</i> . . . . .	69
6.2	Controlo de velocidade . . . . .	73
6.3	Sumário . . . . .	75
<b>7</b>	<b>Resultados</b>	<b>77</b>
7.1	Equipa de dois robôs . . . . .	77

7.1.1	Cenário 1 - O <i>Leader</i> entra numa passagem sem saída . . . . .	78
7.1.2	Cenário 2 - Passagens estreitas e passagem pela porta . . . . .	82
7.2	Equipa de três robôs . . . . .	87
7.2.1	Cenário 1 - Humanos colocam carga nos robôs . . . . .	87
7.2.2	Cenário 2 - Equipa desvia-se de um obstáculo móvel . . . . .	90
7.3	Sumário . . . . .	95
<b>8</b>	<b>Conclusão</b>	<b>97</b>
8.1	Síntese . . . . .	97
8.2	Trabalho futuro . . . . .	99
<b>A</b>	<b>Valores dos parâmetros</b>	<b>101</b>
A.1	Capítulo 3 . . . . .	101
A.2	Capítulo 5 . . . . .	101
A.3	Capítulo 6 . . . . .	101
A.3.1	Equipa de dois robôs . . . . .	102
A.3.2	Equipa de três robôs . . . . .	102
<b>B</b>	<b>Versão digital</b>	<b>103</b>

# Capítulo 1

## Introdução

---

Neste capítulo é apresentada a motivação para a realização desta dissertação e um pequeno estado da arte relativo ao desenvolvimento de arquitecturas de controlo para sistemas multi-robô em que a tarefa consiste no transporte de objectos. O capítulo termina com uma breve descrição de como se encontra organizada esta dissertação.

---

O sonho de uma sociedade automatizada, com máquinas autónomas capazes de executar o trabalho do Homem acompanha-nos desde há alguns séculos. O ser humano procurou sempre compensar as suas fragilidades físicas com a construção de artefactos. Com o passar do tempo estes dispositivos foram evoluindo e o Homem foi criando máquinas capazes de realizar as tarefas mais árduas. Um exemplo de tarefa árdua é o transporte de objectos de grandes dimensões, como ilustra a figura 1.1.

Actualmente investigações no âmbito da robótica e agentes autónomos de uma maneira geral estão em ascensão, demonstrando a importância destas áreas de investigação. O rápido desenvolvimento dos computadores tem elevado de forma significativa o conceito de agentes autónomos. As vantagens dos agentes autónomos são inúmeras, mas a principal é a capacidade de adaptação ao ambiente e de agir sem a intervenção do Homem. Tarefas como a manipulação e/ou transporte de objectos em ambientes radioactivos, em explorações extra-planetárias, em ambientes poluídos com gases nocivos para o Homem,



Figura 1.1: Ilustração de uma possível tarefa árdua.

são uma grande motivação para o uso de robôs móveis autónomos.

Desde finais da década de 80, muitos são os grupos de investigação que se dedicam ao controlo e coordenação de vários robôs móveis autónomos, que em conjunto devem transportar objectos de grandes dimensões. Exemplo de tal, são os trabalhos de: Kosuge *et al.* (2000), Chaimowicz *et al.* (2001), Pereira *et al.* (2002), Zaerpoor *et al.* (2003), Bicho *et al.* (2004), Zaerpoor *et al.* (2005) e Soares *et al.* (2007).

Um dos problemas fundamentais no controlo de equipas de robôs que em conjunto devem transportar um objecto, é movimentarem-se mantendo uma configuração geométrica fixa, pois o objecto que transportam, à partida é rígido, o que faz com que se tenham de movimentar de forma precisa, logo a margem de erro tem que ser mínima (Lewis and Tan (1997), Kosuge *et al.* (2000), Pimentel *et al.* (2002) e Zaerpoor *et al.* (2003)). É de salientar que quanto maior for o número de robôs que constituem a equipa mais complicado se torna o problema, uma vez que os movimentos têm que ser ainda mais precisos. O problema torna-se ainda mais complexo quando os robôs para além de transportarem o objecto terem que simultaneamente evitar colisões com obstáculos.

Os trabalhos realizados podem classificar-se em duas abordagens:

- i) **centralizada** baseia-se na existência de um controlador central, responsável por calcular o deslocamento desejado para o objecto e a força necessária para o esta-

bilizar. Neste tipo de abordagem, todos os robôs têm a tarefa de acompanhar o movimento do objecto e de o estabilizar (Hashimoto *et al.* (1991), Hashimoto *et al.* (1993b), Koga *et al.* (1992)). Nestes sistemas, dentro de um mesmo grupo de robôs, para simplificar o cálculo, há robôs responsáveis pelo movimento estável do objecto e outros responsáveis pela rotação do mesmo, continuando apenas a existir um controlador central para todo o grupo (Nguyen (1986), Ahmadabadi and Nakano (2001)). Neste tipo de abordagem, apesar de existirem robôs com tarefas diferentes, a unidade central continua a ser a mesma para todo o grupo, requerendo assim uma grande largura de banda, quando comparada com a abordagem baseada em comportamentos;

- ii) **baseada em comportamentos** é caracterizada pela não necessidade de modelos dos ambientes em que os robôs operam. Nesta abordagem o sistema é dividido em múltiplos módulos comportamentais, conferindo-lhes uma tarefa diferente a cada um desses módulos comportamentais. Os comportamentos individuais não possuem em si qualquer tipo de inteligência. As acções não são explicitamente planeadas, mas emergem da activação de comportamentos pré-estruturados. Esta abordagem tornou-se numa forte hipótese para a resolução dos problemas inerentes à abordagem centralizada uma vez que:
  - a) não existe separação do sistema em sub-sistema perceptual, central e de actuação;
  - b) não existe uma unidade central de controlo;
  - c) não necessita do modelo do ambiente em que a equipa de robôs navega.

## 1.1 Transporte de objectos por equipas de robôs

### 1.1.1 Soluções para sistemas duo-robô

No trabalho intitulado por “*Transportation of a single object by two decentralized controlled nonholonomic mobile robots*”, de Kosuge *et al.* (1998), os autores propõem um

algoritmo de controlo descentralizado para uma equipa de dois robôs móveis autónomos, segundo uma estratégia *Leader-Follower*. Ao robô que designam por *Leader*, é fornecida a trajectória do objecto, e ao robô *Follower* cabe-lhe a tarefa de ajudar o *Leader*, estimando o movimento do objecto. O robô *Follower* é controlado através da sua direcção de navegação e da sua velocidade na direcção em que navega. Os robôs utilizados são não holonómicos. Cada robô está equipado com um suporte centrado no centro de rotação do respectivo robô, e é constituído por uma junta rotacional livre e por um apoio do objecto, como se pode ver na figura 1.2. Neste trabalho, os autores não têm em consideração a presença de obstáculos, e os robôs apenas se movimentam segundo uma linha recta.

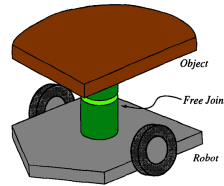


Figura 1.2: Junta rotacional livre. <sup>1</sup>

Este grupo apresenta uma solução para o problema de transportar um objecto e simultaneamente evitar colisões com obstáculos na publicação intitulada por “*Decentralized motion control of two tracked mobile robots transporting a single object in coordination based on function allocation concept*”, de Kosuge *et al.* (2000). Neste trabalho é proposta uma estratégia de controlo descentralizado, baseado num conceito de alocação de funções. A presença de obstáculos é tida em conta pelos robôs, ou seja, os robôs para além de terem que transportar o objecto têm que simultaneamente evitar colisões com obstáculos. Tal como no trabalho de (Kosuge *et al.*, 1998), é seguida uma estratégia *Leader-Follower*. O suporte do objecto é o mesmo. A posição do robô *Follower* é controlada de forma que este alinhe a sua orientação com a direcção do movimento do objecto. Para evitar colisões com os obstáculos, é assumido que tem que existir uma distância mínima entre o centro do objecto e um determinado obstáculo, o que implica a necessidade de conhecer previamente a localização dos obstáculos. É de salientar que os autores não referem, e não é óbvio, como é calculada essa distância.

<sup>1</sup>Figura retirada de Kosuge *et al.* (1998).



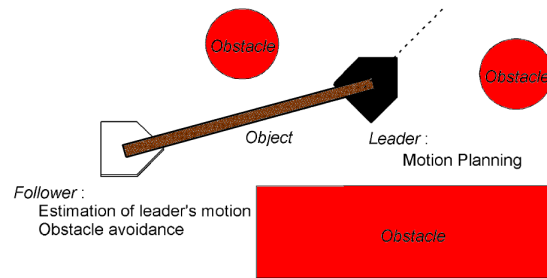


Figura 1.3: Conceito de alocação de funções. <sup>2</sup>

No trabalho “*Cooperative transportation by two four-legged robots with implicit communication*” de Aiyama *et al.* (1999), é abordado o transporte de objectos por dois robôs móveis com pernas. É usada uma estratégia em que não existe nenhum robô *Leader* e também não existe comunicação explícita entre ambos. O objecto é apoiado num sensor de força de seis eixos (ver figura 1.4), e todo o controlo é baseado na informação sensorial retirada do mesmo. É assumido que a configuração de transporte é fixa, isto é, um à frente e outro a trás, sendo também a direcção de navegação fixa. A direcção de navegação é fornecida apenas ao robô da frente. Se através da informação do sensor de força do suporte o robô da frente sentir que o robô de trás está atrasado ou adiantado, então o da frente terá de esperar ou acelerar, respectivamente. Se através da informação do sensor de força do suporte o robô de trás sentir que o robô da frente está longe ou perto então o robô de trás terá que acelerar ou abrandar (ou se necessário parar), respectivamente. Neste trabalho não é tido em consideração a presença de obstáculos, e os robôs apenas se movem em linha recta.

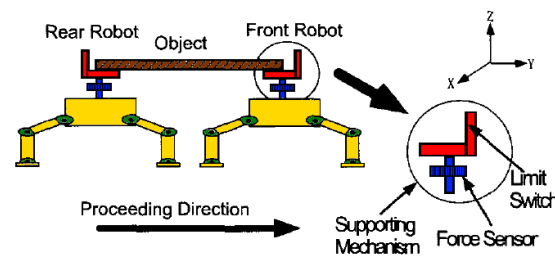


Figura 1.4: Mecanismo de suporte dos robôs utilizados. <sup>3</sup>

Nos trabalhos intitulados por “*Coordination of multiple mobile robots in an object car-*

<sup>2</sup>Figura retirada de Kosuge *et al.* (2000).

<sup>3</sup>Figura retirada de Aiyama *et al.* (1999).

*rying task using implicit communication*” de Pereira *et al.* (2002), e por “*On the development of cooperative behavior-based mobile manipulators*” de Pimentel *et al.* (2002) os autores propõem o controlo de robôs móveis para o transporte de objectos segundo comportamentos básicos. Também é adoptada uma estratégia *Leader-Follower*, em que ao robô *Leader* cabe-lhe a tarefa de dar rumo ao objecto a transportar. Para facilitar a tarefa de evitar obstáculos é usada a troca de liderança que é efectuada através de comunicação implícita, a qual tem como vantagem o uso apenas da informação sensorial de cada robô. Os robôs possuem um suporte para o objecto que é sensível à força. Os robôs têm grande dificuldade em detectar os obstáculos e evitá-los, devido a possuírem manipuladores. Em situações de presença de obstáculos os robôs devem recuar (ver figura 1.5(a)), nestas condições, o robô *Leader* pode ceder a liderança da equipa por breves instantes ao robô *Follower*. Cada robô tem dois modos básicos de controlo, isto é, o de liderar e o de seguir. O robô que lidera navega numa determinada direcção enquanto o outro tenta manter as forças aplicadas ao objecto constantes. Inicialmente, o objecto está em equilíbrio uma vez que a soma das forças aplicadas ao objecto são nulas. A troca de liderança é feita sempre que o *Follower* detecte que o *Leader* recua e sem recurso a comunicação explícita. A navegação da equipa assenta em cinco comportamentos básicos: *Function*, *World*, *Goal*, *Avoid* e *Grasp*. O *Function* é responsável pela deliberação do estado principal do robô (liderar ou seguir). O *World* é responsável por “observar” o ambiente em que o robô navega por forma a activar o comportamento *Goal* ou o *Avoid*. O *Goal* é responsável pelo movimento na direcção do alvo. O *Avoid* é responsável pelo movimento que evite os obstáculos detectados. O *Grasp* controla a posição do robô de forma que este aplique sempre uma força apropriada ao objecto. É de salientar que quando o robô *Follower* recua devido à troca de liderança, este não tem em consideração a existência ou não de obstáculos atrás de si, o que pode levar à colisão. Dado o facto do robô *Follower* recuar numa trajectória espelhada à do robô *Leader* faz com que seja necessário memorizar a trajectória por parte do robô *Follower*.

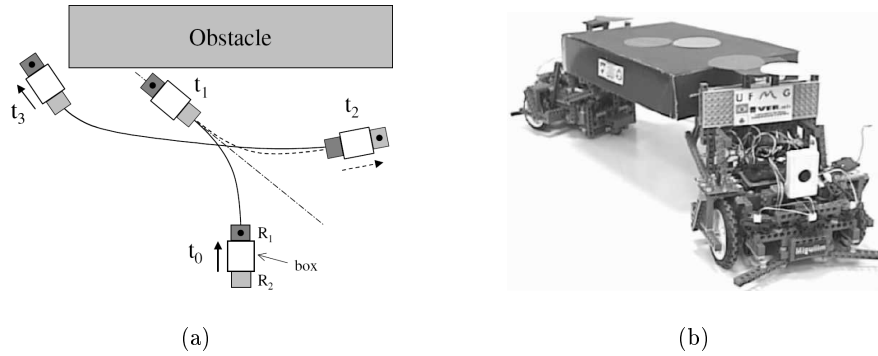


Figura 1.5: Na imagem (a) a linha a tracejado mostra a trajectória do *Follower* ( $R_2$ ) quando este lidera o grupo, e a linha sólida quando não lidera. O robô com o ponto preto é o que no momento conduz o grupo. No instante  $t_1$  o *Leader* cede a liderança, e no  $t_2$  o *Follower* devolve-a. A figura mostra a trajectória espelhada determinada heurísticamente pelo *Follower* para ajudar o grupo a evitar o obstáculo. Na imagem (b) pode ver-se os robôs que o grupo utilizou. <sup>4</sup>

No trabalho intitulado por “*Graspless coordinated transportation over natural flat terrain*” de Velasquez *et al.* (2004) é apresentado um método que coordena as acções de uma equipa de dois robôs, cuja tarefa consiste no transporte de um objecto de massa uniforme em terrenos relativamente planos. É de salientar que todos os outros trabalhos já referidos também têm esta limitação. A estratégia de controlo utilizada também é distribuída. A velocidade de cada robô é ajustada de acordo com o deslocamento que o objecto tem no respectivo suporte, tornando o controlo distribuído sem necessidade de comunicação explícita entre os robôs. Cada robô possui um braço cuja amplitude varia entre uma posição máxima e uma mínima. A posição do objecto relativamente ao robô é medida através do braço, sendo a posição de repouso do braço definida como posição alvo. Cada robô movimenta-se com uma velocidade cuja magnitude varia proporcionalmente com a abertura do braço. Inicialmente os robôs são posicionados à mesma distância do centro de massa do objecto. Apesar dos robôs terem em consideração a existência de obstáculos no ambiente que os rodeia, estes têm de ter uma determinada forma, nunca sendo referida. Esta restrição faz com que o ambiente tenha que ser estruturado ou até mesmo os robôs terem que conhecer, *a priori*, o ambiente em que navegam.

<sup>4</sup>Figuras retiradas de Pereira *et al.* (2002) e de Pimentel *et al.* (2002).

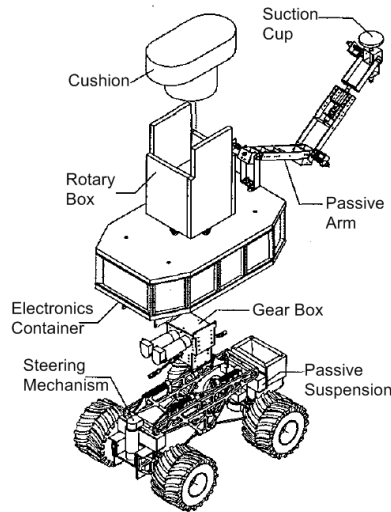


Figura 1.6: Componentes do robô. <sup>5</sup>

### 1.1.2 Soluções para sistemas multi-robô

No trabalho intitulado por “*Coordinative object-transportation by multiple industrial mobile robots using coupler with mechanical compliance*” de Hashimoto *et al.* (1993a) é proposta uma abordagem para coordenar o movimento de robôs móveis industriais, em que a tarefa que lhes é incumbida é o transporte cooperativo de um objecto. O sistema desenvolvido é designado por *METROS (Multiple Transfer Robots System)*. O movimento de cada robô é efectuado por meio de duas rodas diferenciais. O mecanismo de acoplamento é ligado a uma paleta para permitir o transporte de objectos de formas variadas (ver figura 1.7(a)). O mecanismo referido sofreu posteriormente uma alteração para permitir maior flexibilidade aos robôs passando assim os robôs a dignarem-se de robôs manipuladores. Esta alteração do mecanismo de acoplamento foi apresentada no trabalho intitulado por “*Object-transportation control by multiple wheeled vehicle-planar cartesian manipulator systems*” (Hashimoto *et al.*, 1995). O movimento do objecto é controlado pela força resultante produzida pelos robôs. O mecanismo de acoplamento é constituído por uma junta prismática passiva com dois graus de liberdade, uma junta rotacional com um grau de liberdade e por um suporte onde assenta a paleta (ver figura 1.7(b)). A flexibilidade do mecanismo de acoplamento é implementada com a

<sup>5</sup>Figura retirada de Velasquez *et al.* (2004).

introdução de molas nas juntas prismáticas, as quais podem expandir-se e retrair-se ao longo de cada uma das juntas. A junta rotacional permite que o robô rode sobre si próprio livremente em qualquer direcção. O sistema é constituído por um robô designado de *Leader* e por mais três robôs designados por *Helpers*. O robô *Leader* define o comportamento do objecto e comunica-o a todos os robôs *Helpers*. Por sua vez, os robôs *Helpers* com a informação fornecida pelo *Leader* tentam concretizar a tarefa pretendida. A informação fornecida pelo *Leader* aos robôs *Helpers* sobre o comportamento do objecto resume-se à força resultante a aplicar ao objecto. Esta informação é distribuída equitativamente por todos os robôs *Helpers*, que com esta informação calculam a posição e orientação desejadas em relação ao objecto. Para evitar os erros devido às constantes das molas, o valor do sensor de força do mecanismo de acoplamento é realimentado para o cálculo da força e orientação desejadas de cada robô. Este trabalho não aborda o problema de evitar colisões com obstáculos.

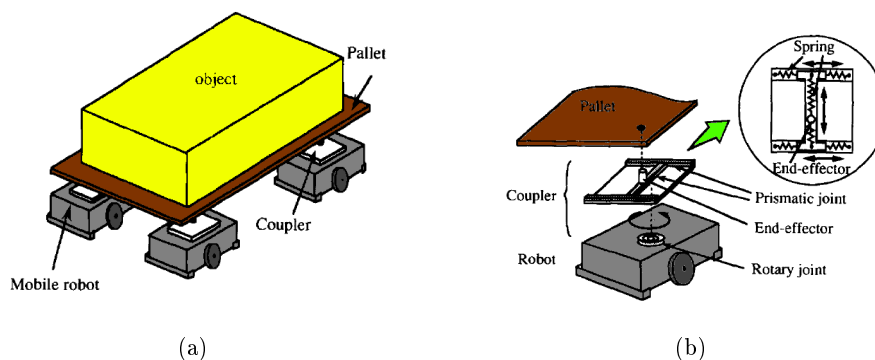


Figura 1.7: Sistema *METROS*.<sup>6</sup>

No trabalho intitulado por “*High precision formation control of multiple robots using virtual structures*” de Lewis and Tan (1997) é abordado o problema da manutenção de uma configuração geométrica durante o movimento de uma equipa de robôs. Aqui é considerado que os robôs são parte integrante de uma estrutura virtual, como se fossem partículas embutidas nessa estrutura. As restrições impostas mecanicamente por uma estrutura real, na equipa de robôs são impostas pela estratégia de controlo. A solução apresentada é capaz de gerar movimentos para os robôs e simultaneamente satisfazer duas condições: *i*) progressão numa dada direcção; *ii*) manutenção de uma rígida rela-

<sup>6</sup>Figuras retiradas de Hashimoto *et al.* (1993a).

ção geométrica entre si. A estrutura virtual move-se para acompanhar os movimentos dos robôs e os robôs movem-se para acompanhar o movimento da estrutura virtual. Este facto realça o controlo bidireccional desta solução. O algoritmo de controlo é composto por quatro etapas: *i)* alinhar a estrutura virtual com a posição actual dos robôs; *ii)* mover a estrutura virtual em translação e/ou rotação; *iii)* calcular as trajectórias individuais de cada robô para que estes se movam para as direcções desejadas; *iv)* ajustar as velocidades desejadas dos robôs para que estes consigam percorrer a trajectória desejada. Os requisitos de comunicação dependem de três factores: *i)* do número de robôs na formação; *ii)* dos bits necessários para codificar a posição de cada robô; *iii)* o período de actualização do ciclo de controlo dos robôs. Esta solução assenta num sistema centralizado, o que como é evidente, está sujeito às desvantagens do uso de sistemas centralizados, como por exemplo, falhas por parte da unidade central. Neste trabalho não é abordada a questão de evitar colisões com obstáculos.

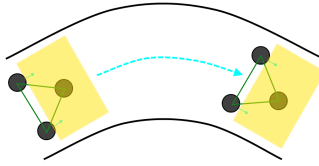


Figura 1.8: Robôs móveis transportando uma caixa mantendo uma configuração geométrica fixa com a caixa, para assegurar o sucesso da tarefa. <sup>7</sup>

Nos trabalhos de Ahmadabadi and Nakano (2001), Zaerpoor *et al.* (2003) e Zaerpoor *et al.* (2005), é utilizada uma estratégia de controlo distribuída para a coordenação do movimento de equipas de robôs móveis que em conjunto devem transportar um objecto. A estratégia utilizada é designada pelos autores de *Constrain and Move*. Segundo esta estratégia, para rodar um objecto em torno de um ponto fixo os robôs têm que restringir o movimento do objecto nas direcções  $x$  e  $y$  e produzir um binário adequado em torno do eixo de rotação do objecto. Para manterem o objecto estável no ponto de contacto Robô-Objecto, este tem que se manter sempre dentro do cone de fricção e o robô não pode puxar o objecto na sua direcção. Para estabilizar o ponto de contacto Robô-Objecto são necessários os comportamentos: *i)* não puxar o objecto; *ii)* a junta Base Móvel-Braço deve produzir um binário nulo; *iii)* o objecto deve ser mantido na zona

<sup>7</sup>Figura retirada de Lewis and Tan (1997).

de fricção. Para manter as restrições ao movimento do objecto quando o mesmo roda em torno de um ponto fixo, os robôs têm de controlar o seu ângulo Objecto-Braço. Para imobilizar o centro de rotação do objecto são necessários pelo menos três robôs e para restringir o movimento do objecto, as direcções dos braços dos robôs têm que se interceptar. Para concretizar estas restrições são necessários os comportamentos: *i)* não movimentar a junta Braço-Base Móvel na direcção do braço; *ii)* manter o ângulo Robô-Objecto no seu valor inicial; *iii)* seguir o objecto numa direcção perpendicular ao braço. Os robôs responsáveis por restringir o movimento do objecto são posicionados de tal forma que o seu braço esteja numa direcção perpendicular ao percurso do objecto. Cada robô movimenta a sua base móvel de forma que o seu ponto de contacto com o objecto não se mova na direcção do seu braço. Para concretizar estas restrições são necessários os comportamentos: *i)* não puxar o objecto; *ii)* conduzir o ponto de contacto Robô-Objecto para o percurso definido; *iii)* manter o ângulo Robô-Objecto dentro de um determinado valor de referência. Quando num determinado instante o objecto não se encontra no percurso desejado, a velocidade desejada do objecto é recalculada de forma a eliminar o erro existente. A função designada por percurso desejado retorna a posição desejada do objecto em cada instante. Assim sendo o erro é a diferença entre o valor retornado e a actual posição do objecto. Para compensar este erro a velocidade é recalculada de forma proporcional ao erro calculado. No trabalho desenvolvido por este grupo a trajectória desejada é fornecida aos robôs e o problema de evitar colisões com obstáculos não é abordado.

No trabalho intitulado “*Decentralized control of multiple mobile manipulators handling a single object in coordination*” de Kume *et al.* (2002) é proposto um algoritmo de controlo descentralizado para uma equipa constituída por vários manipuladores móveis que cooperativamente devem transportar um objecto. O algoritmo proposto é do tipo *Leader-Follower*, mas segundo o pressuposto de não utilizar as relações geométricas entre os robôs que maioritariamente este tipo de algoritmos utiliza. Quando é fornecido um comando de movimento ao robô *Leader* este aplica uma força a todos os outros robôs através do objecto. Os *casters* virtuais dos *Followers* direccionam-se na direcção do robô *Leader*. Podem ser executados três tipos de movimentos pelos *casters* virtuais: *i)* movimento de translação da roda segundo a direcção de navegação da roda; *ii)*

movimento rotacional da roda em torno do eixo rotacional; *iii*) movimento rotacional da junta esférica em torno de si própria. O problema de evitar colisões com obstáculos não é abordado neste trabalho.

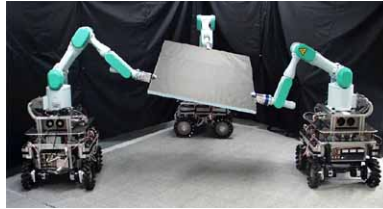


Figura 1.9: Sistema experimental dos robôs móveis com manipuladores. <sup>8</sup>

O grupo de Chaimowicz *et al.* (2001), no trabalho intitulado por “*An architecture for tightly coupled multi-robot cooperation*”, estenderam o seu trabalho de sistemas duo-robô (Pereira *et al.*, 2002) a sistemas multi-robô. A arquitectura segue uma estratégia *Leader-Followers*. Também neste trabalho existem dois métodos de troca de liderança, a requisição e a rejeição. Um robô requisita a liderança quando não é capaz de seguir o planeamento efectuado pelo robô Leader ou quando a execução da tarefa é mais fácil para o requisitante. A rejeição acontece quando o robô Leader não é capaz de concluir a tarefa ou quando a sua liderança numa determinada tarefa termina. Com o intuito de dotar os robôs de maior eficiência, este grupo de trabalho propôs uma metodologia para execução de tarefas cooperativas. Nesse trabalho, intitulado por “*Dynamic role assignment for cooperative robots*” de Chaimowicz *et al.* (2002) o controlo é baseado na atribuição dinâmica das tarefas na qual os robôs são capazes de trocar entre si a execução das mesmas durante a cooperação. Assumindo dinamicamente as suas tarefas e trocando entre si a execução de tarefas os robôs são assim capazes de executar o objectivo final de forma mais eficiente. As tarefas foram definidas como sendo: *i*) *Allocation* - o robô assume uma nova tarefa depois de ter concluído a execução de outra; *ii*) *Reallocation* - o robô interrompe a tarefa que está a executar para começar ou continuar outra tarefa; *iii*) *Exchange* - dois robôs podem trocar entre si a realização de tarefas. No trabalho realizado por este grupo, é sempre assumido que existe um planeamento prévio das trajectórias e não é claro como é que é abordado o problema de evitar colisões com obstáculos.

<sup>8</sup>Figura retirada de Kume *et al.* (2002).



## 1.2 Objectivos da dissertação

A motivação para o uso de sistemas dinâmicos não lineares provém de trabalhos recentes que demonstram que os métodos da teoria de sistemas dinâmicos não lineares, podem ser usados como uma linguagem conceptual e ferramenta para descrever o acoplamento dinâmico entre um robô e o ambiente que o rodeia, incluindo os seus *partners* (ex: Bicho and Schöner (1997), Bicho *et al.* (2000), Bicho *et al.* (2003), Bicho *et al.* (2004) e Soares *et al.* (2007)). O trabalho proposto nesta dissertação tem como objectivo a implementação e validação de uma nova estratégia de controlo, baseada em sistemas dinâmicos não lineares, para uma equipa de robôs móveis autónomos que em conjunto devem transportar um objecto desde uma posição inicial até uma posição de destino, e em simultâneo evitar colisões com obstáculos estáticos e/ou dinâmicos que possam encontrar no seu caminho. O objecto pode ser de várias formas e a equipa pode ser constituída por dois ou três robôs, ver figura 1.10. Esta nova arquitectura de controlo, formalizada como sistemas dinâmicos não lineares, foi desenvolvida no âmbito dos trabalhos de doutoramento de Soares (2007).

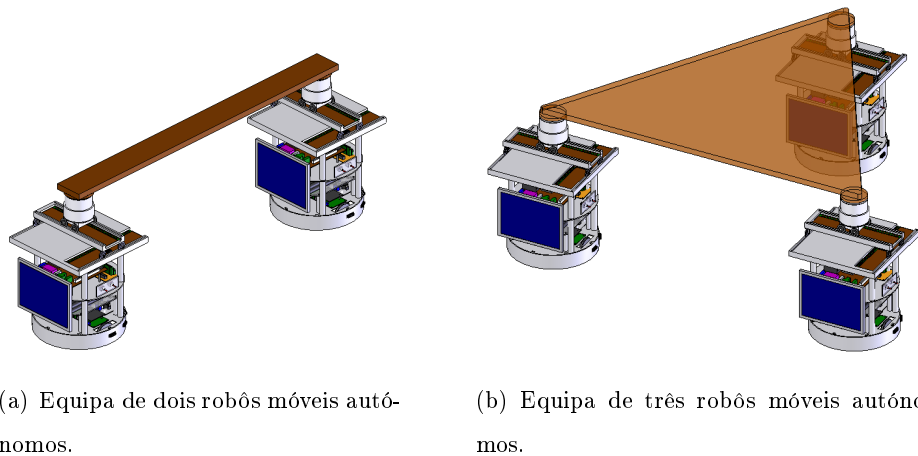


Figura 1.10: Equipas de dois e três robôs móveis autónomos.

Os principais objectivos a atingir nesta dissertação foram:

- o projecto de um novo suporte para apoiar o objecto a transportar que permita detectar em que direcção o grupo se encontra relativamente ao próprio robô, e o

quão desviado do seu centro o objecto está em cada instante;

- implementação, teste e validação da arquitectura de controlo desenvolvida em ambiente de simulação por Soares (2007).

### 1.3 Estrutura da dissertação

A restante dissertação encontra-se organizada da seguinte forma:

- No capítulo 2 são apresentados os princípios básicos para a geração de comportamentos baseados em sistemas dinâmicos não lineares;
- No capítulo 3 são apresentadas as equipas de robôs, as suas tarefas, e a informação a comunicar entre os robôs;
- No capítulo 4 é apresentado o projecto do novo suporte para o apoio do objecto a transportar;
- No capítulo 5 é apresentado todo o controlo que governa o movimento do robô *Leader*;
- No capítulo 6 é apresentado o controlo que governa o movimento do(s) robô(s) *Helper(s)* numa equipa constituída por dois ou por três robôs móveis autónomos, seguindo a filosofia *Leader - Helper(s)*;
- No capítulo 7 são apresentados os resultados obtidos pela implementação da arquitectura de controlo numa equipa de dois e numa de três robôs móveis autónomos;
- Por fim, no capítulo 8 é feita uma discussão dos resultados obtidos e descritas as perspectivas quanto ao trabalho futuro.

## Capítulo 2

# Princípios básicos para a geração de comportamentos usando sistemas dinâmicos não lineares

---

A abordagem dinâmica para a geração de comportamentos originalmente proposta em Schöner *et al.* (1995), fornece um conjunto de conceitos e princípios baseados na teoria dos sistemas dinâmicos. Estes conceitos são utilizados como linguagem teórica, a partir da qual podem ser construídas arquitecturas de controlo. Neste capítulo são apresentados os conceitos e os princípios básicos da abordagem dinâmica, de forma similar ao apresentado por Bicho (2000).

---

A abordagem dinâmica assenta nos seguintes princípios:

- Conceito de variáveis comportamentais: as quais descrevem um comportamento particular e definem dimensões particulares nas quais o comportamento pode variar;
- Conceito de dinâmica comportamental: de acordo com a qual é possível gerar

comportamentos através de soluções atractoras de sistemas dinâmicos.

Estes conceitos foram extendidos e aplicados por Bicho (2000) para desenvolver arquitecturas de controlo de robôs móveis autónomos, tornando-os capazes de se movimentarem em ambientes desconhecidos *a priori*.

## 2.1 Variáveis comportamentais

Para descrever um comportamento, no que se refere à abordagem dinâmica, é necessário definir as variáveis que permitem descrever, parametrizar e internamente representar o estado do sistema, isto é, o comportamento. Num dado instante o comportamento corresponde a um ponto no espaço da dimensão comportamental. As variáveis escolhidas devem satisfazer os requisitos seguintes:

- As variáveis comportamentais devem permitir a implementação de sistemas de controlo que imponham os seus valores num sistema de actuadores;
- Possibilidade de especificar os valores obtidos pelos sensores da plataforma robótica;
- Os valores especificados para uma variável comportamental, que representam uma tarefa, devem ser independentes do seu valor actual;
- Em qualquer instante, um comportamento tem que estar associado a um valor específico da correspondente variável comportamental e a cada tarefa tem que estar associado um valor ou conjunto de valores desta variável.

Um exemplo de variável comportamental é a direcção de navegação de um robô móvel autónomo cujo movimento deve ser controlado de forma que direcções onde se encontrem obstáculos sejam evitadas, enquanto que a direcção onde se encontra o alvo seja alcançada. Para exprimir o comportamento que este movimento representa no ambiente que rodeia o robô móvel autónomo (relativamente a um eixo arbitrário mas fixo,  $x$ ), a

direcção de navegação,  $\phi$ , é uma variável comportamental adequada. Na figura 2.1  $\psi_{obs}$  é um valor a evitar enquanto que  $\psi_{tar}$  é um valor desejado, isto é, a alcançar.

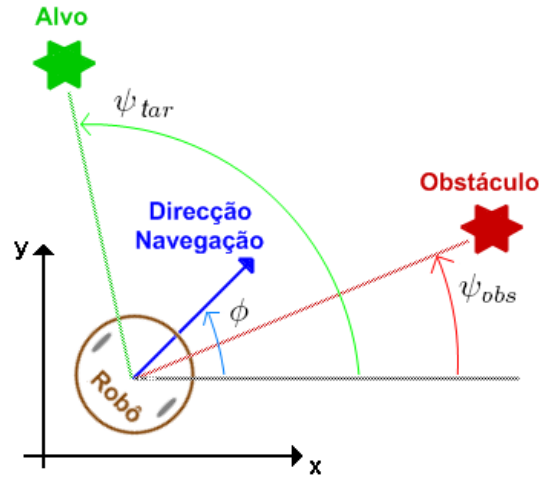


Figura 2.1:  $\phi$  é a variável comportamental escolhida para gerar o movimento do robô móvel autónomo na direcção do alvo e simultaneamente evitar colisões com obstáculos.

As velocidades linear e angular são também possíveis variáveis comportamentais.

## 2.2 Dinâmica comportamental

Torna-se necessário gerar valores ao longo do tempo para as variáveis comportamentais, isto é, variáveis que se querem controlar, as quais controlam o comportamento do robô. Para alcançar este objectivo é implementado um sistema dinâmico para as variáveis comportamentais. Os sistemas dinâmicos são contínuos no tempo e são definidos por uma equação diferencial na qual as variáveis de estado são variáveis comportamentais. Por exemplo, para a direcção de navegação,  $\phi$ , o sistema dinâmico define a taxa de variação da direcção de navegação em função do seu valor actual, isto é:

$$\frac{d\phi(t)}{dt} = f(\phi(t), \text{parâmetros}) \quad (2.1)$$

A equação 2.1 define um campo vectorial, isto é, a cada ponto do espaço é associado um vector  $f(\phi)$ . Cada um destes vectores determina a direcção na qual o sistema se

moverá e a sua taxa de variação com a qual se moverá da direcção em que se encontra.

## 2.3 Pontos de equilíbrio

Como princípio de *design*, pretende-se soluções particulares dos sistemas dinâmicos designadas por pontos de equilíbrio ou pontos fixos. Estas soluções são os pontos nos quais o campo vectorial é nulo.

$$\left. \frac{d\phi(t)}{dt} \right|_{\phi=\phi_{\text{PontoEquilíbrio}}} = f(\phi_{\text{PontoEquilíbrio}}) = 0 \quad (2.2)$$

Pontos de equilíbrio são soluções constantes dos sistemas dinâmicos, isto é, em que o sistema não muda de estado. Mas dizer que um sistema está num ponto de equilíbrio, não significa que ele é estável, pois este pode ser de duas naturezas: estável ou instável. Se o sistema estiver num ponto de equilíbrio instável, qualquer perturbação que o sistema sofra, este sairá desse ponto de equilíbrio instável e irá, se nada o contrariar, para um ponto de equilíbrio estável, como se pode ver na figura 2.2.

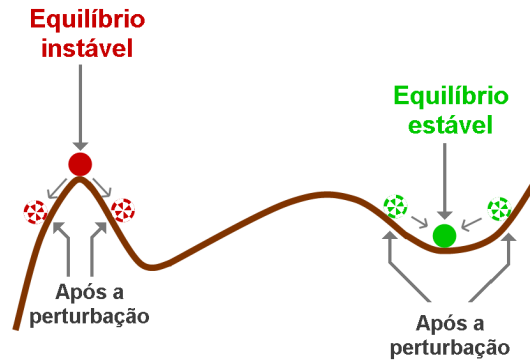


Figura 2.2: Pontos de equilíbrio estável e instável. Inicialmente as duas bolas estão em repouso, após ser aplicada uma pequena perturbação em cada uma das bolas, estas comportam-se de forma diferente. A bola da esquerda, bola vermelha, devido à perturbação cai para um dos lados, direito ou esquerdo, enquanto que a bola da direita, bola verde, oscila em torno do ponto em que estava, mas ao fim de algum tempo retorna ao ponto de equilíbrio. No caso da bola do lado esquerdo, bola vermelha, o ponto de equilíbrio designa-se de instável e no caso da bola da direita, bola verde, o ponto de equilíbrio designa-se de assintoticamente estável.

Para um sistema dinâmico de uma variável a estabilidade do ponto de equilíbrio pode

facilmente ser medido graficamente, ver figura 2.3.

Os gráficos (a) da figura 2.3 representa um sistema dinâmico não linear, (b) um sistema dinâmico linear, (c) um sistema dinâmico não linear e (d) um sistema dinâmico linear. Os sistemas de (a) e de (b) exibem um ponto de equilíbrio em  $\phi = \phi_A$  e  $\phi = \phi_B$ , respectivamente. Dado o facto do declive nestes pontos ser negativo, os pontos de equilíbrio são designados de pontos de equilíbrio assintoticamente estáveis, pelo que são designados também por atractores, porque atraem a variável comportamental para o valor por si especificado. Para verificar isto, basta considerar um ponto ligeiramente à direita de  $\phi_A$  ou  $\phi_B$ , por exemplo  $\phi_{A1}$  ou  $\phi_{B1}$ , respectivamente. Neste caso, a taxa de variação da variável comportamental é negativa o sistema adquire valores cada vez mais próximos do ponto de equilíbrio  $\phi = \phi_A$  ou  $\phi = \phi_B$ , respectivamente. Analogamente para valores ligeiramente à esquerda de  $\phi_A$  ou  $\phi_B$ , por exemplo  $\phi_{A2}$  ou  $\phi_{B2}$ , respectivamente, a taxa de variação da variável comportamental é positiva, pelo que esta assume valores cada vez mais próximos do ponto de equilíbrio  $\phi = \phi_A$  ou  $\phi = \phi_B$ , respectivamente. Os campos vectoriais de ambos os sistemas dinâmicos comportam-se como forças atractivas que conduzem ao estado especificado por  $\phi = \phi_A$  ou  $\phi = \phi_B$ , respectivamente. Por exemplo, para o comportamento *dirigir-se para o alvo*, a direcção na qual está localizado o alvo é um valor desejado para a direcção de navegação do robô, pode-se tornar esta direcção num atractor erigindo um campo vectorial com um zero nessa direcção e com um declive negativo. O alcance das variáveis comportamentais segundo o qual é exercida a força atractiva pode ser limitada, gráfico 2.3(a), ou ilimitada, gráfico 2.3(b). Por isso, um conceito importante associado com os pontos de equilíbrio é a bacia de atracção. Para um dado atractor, esta bacia de atracção diz respeito à região do espaço na qual todas as condições iniciais são abrangidas pelo atractor. Contrariamente, quando o declive num ponto de equilíbrio é positivo, como é o caso dos gráficos 2.3(c) e 2.3(d), o ponto de equilíbrio é um ponto instável, e por isso designado de repulsor, um vez que repele o sistema do seu valor. Para verificar isto, basta considerar um ponto ligeiramente à direita de  $\phi_C$  ou  $\phi_D$ , por exemplo  $\phi_{C1}$  ou  $\phi_{D1}$ , respectivamente. Neste caso, como a taxa de variação da variável comportamental é positiva o sistema adquire valores cada vez mais afastados do ponto de equilíbrio. Analogamente para valores ligeiramente à esquerda de  $\phi_C$  ou  $\phi_D$ , por exemplo  $\phi_{C2}$  ou  $\phi_{D2}$ , respectivamente, a taxa de variação da variável

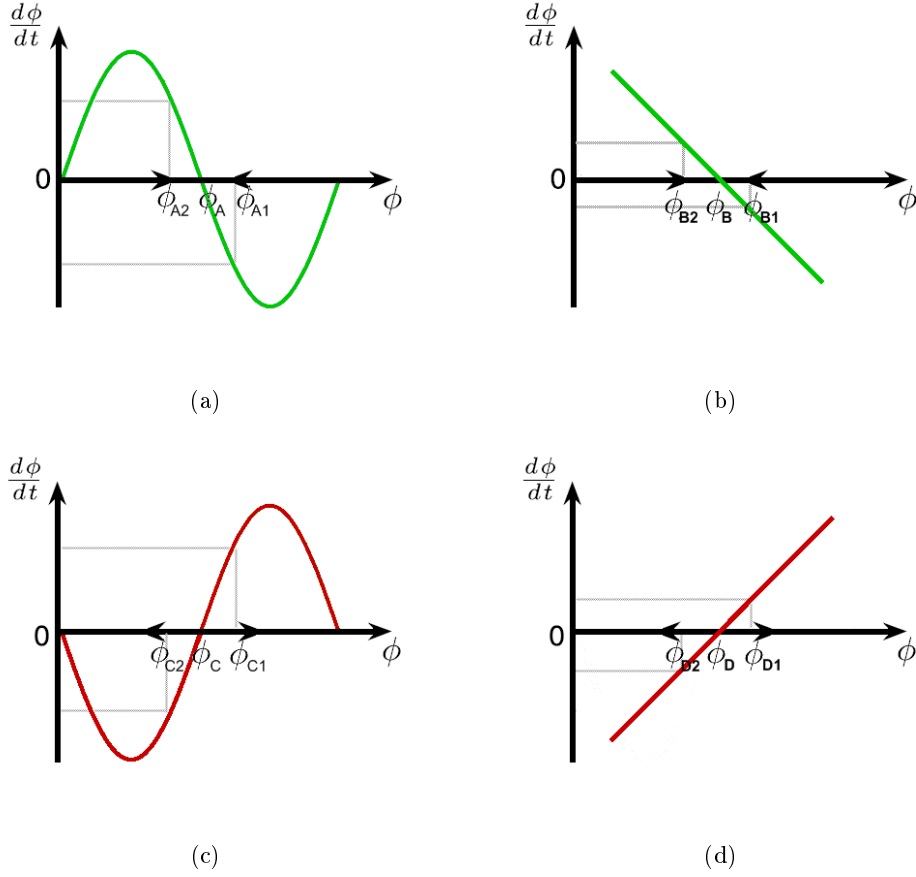


Figura 2.3: Quatro gráficos de sistemas dinâmicos de uma variável, pois a taxa de variação de  $\frac{d\phi}{dt}$  está definida em função de  $\phi$ . Os pontos nos gráficos onde  $\frac{d\phi}{dt} = 0$ , isto é,  $\phi_A$ ,  $\phi_B$ ,  $\phi_C$  e  $\phi_D$ , são pontos de equilíbrio do sistema dinâmico e o declive de  $\frac{d\phi}{dt}$  indica a sua natureza. (a): sistema dinâmico não linear com um ponto de equilíbrio em  $\phi = \phi_A$ , o declive neste ponto é negativo, o que faz com que este ponto de equilíbrio seja um ponto atrator. O sistema converge no tempo para o estado definido pelo ponto atrator. (b): sistema dinâmico linear com um ponto de equilíbrio em  $\phi = \phi_B$ . Tal como no sistema dinâmico de (a) também este converge, neste caso, para  $\phi = \phi_B$ . (c): sistema dinâmico é não linear com um ponto de equilíbrio em  $\phi = \phi_C$ , o declive neste ponto é positivo, o que faz com que o ponto de equilíbrio seja um ponto repulsor. O sistema diverge no tempo do estado definido pelo ponto repulsor. (d): sistema dinâmico linear com um ponto repulsor em  $\phi = \phi_D$ . Tal como no gráfico (c) os sistema diverge deste ponto repulsor. A direcção das setas indica a evolução no tempo da variável comportamental.



comportamental é negativa, pelo que a variável comportamental assume valores cada vez mais afastados do ponto de equilíbrio. Neste caso o campo vectorial erige uma força repulsiva. No ponto de equilíbrio a taxa de variação da variável comportamental é nula, mas basta uma pequena perturbação para que o sistema se afaste deste ponto. Como exemplo, um obstáculo pode ser modelado erigindo uma força repulsiva na direcção em que este está localizado, e assim evitar que a direcção de navegação tome essa direcção em que se encontra(m) o(s) obstáculo(s). A bacia de repulsão, isto é, a região no espaço na qual se fazem sentir os efeitos de repulsão podem ser limitados, gráfico 2.3(c), ou ilimitados, gráfico 2.3(d).

## 2.4 Força dos pontos de equilíbrio

Para os sistemas dinâmicos lineares, gráficos 2.3(b) e 2.3(d), o declive determina a força atractiva ou repulsiva de um ponto de equilíbrio. No caso de um ponto de equilíbrio assintoticamente estável, isto é, atractor, quanto maior for o declive nesse ponto mais forte será a força de atracção exercida por esse ponto e mais rápido o sistema relaxará para o ponto de equilíbrio atractor depois de uma perturbação. De igual forma para um ponto de equilíbrio instável, isto é, repulsor, quanto maior for o declive nesse ponto maior será a força de repulsão exercida por esse ponto e mais rápido o sistema se afasta do ponto de equilíbrio instável após uma perturbação. Por isso o declive representa a estabilidade do sistema nos pontos de equilíbrio. Dado que a relaxação é exponencial pode ser caracterizada por uma taxa de relaxação. Por exemplo, se uma perturbação inicial coloca o sistema dinâmico descrito no gráfico 2.3(b) no ponto equilíbrio  $\phi_B$ , então o sistema evolui no tempo de acordo com a seguinte solução:

$$\phi(t) = \phi_B + (\phi_{B1} - \phi_B) e^{-\frac{t}{\tau}} \quad (2.3)$$

onde  $\tau$  determina o tempo de relaxação com o qual o sistema se aproxima do ponto de equilíbrio  $\phi_B$ . Quando o sistema atinge uma distância  $|\phi_{B1} - \phi_B|$  relativamente ao atractor, o sistema reduz esta distância através de um factor exponencial num intervalo

de tempo  $\tau$ . A relaxação será tanto mais rápida, quanto menor for o valor desse intervalo de tempo  $\tau$ . Por isso,  $\tau$  pode ser usado para caracterizar quantitativamente a estabilidade do sistema nos pontos fixos e pode ser obtido através do inverso do declive do campo vectorial no ponto fixo.

$$\tau = - \left[ \frac{d\phi(t)}{dt} \Big|_{\phi=\phi_{\text{PontoEquilíbrio}}} \right]^{-1} \quad (2.4)$$

Valores negativos de  $\tau$  indicam que o respectivo ponto de equilíbrio é repulsor.  $\tau = 0$  indica que o ponto de equilíbrio é semi-estável. Para sistemas dinâmicos não lineares, gráficos 2.3(a) e 2.3(c), com pontos de equilíbrio hiperbólicos<sup>1</sup> pode-se usar o método de linearização para caracterizar a estabilidade dos pontos de equilíbrio (Perko (1991) e Crawford (1991)). Por exemplo, a estabilidade do atrator  $\phi_A$  do sistema dinâmico não linear descrito na figura 2.3(a) pode ser obtido aproximando este sistema dinâmico a um sistema dinâmico linear na vizinhança do ponto de equilíbrio  $\phi_A$ . Expandindo o campo vectorial  $f(\phi)$  numa série de Taylor em torno do ponto de equilíbrio e mantendo apenas os termos de primeira ordem é produzida a dinâmica com um campo vectorial linear,

$$\frac{d\phi(t)}{dt} = f(\phi) \approx \left( \frac{df(\phi)}{dt} \Big|_{\phi=\phi_A} \right) (\phi - \phi_A) \quad (2.5)$$

cuja solução tem a forma da equação 2.3 e onde a constante de tempo,  $\tau$ , é dada pelo inverso do declive do campo vectorial no ponto de equilíbrio, tal como para um sistema dinâmico linear. Assim sendo, a constante de tempo,  $\tau$ , dada pela equação 2.4, pode também ser usada para calcular a estabilidade dos pontos de equilíbrio de um sistema dinâmico não linear. Fortes variáveis comportamentais têm tempos de relaxação muito curtos.

---

<sup>1</sup>Os pontos de equilíbrio são designados por hiperbólicos quando não têm valor próprio no eixo imaginário.

## Capítulo 3

# Robôs móveis

---

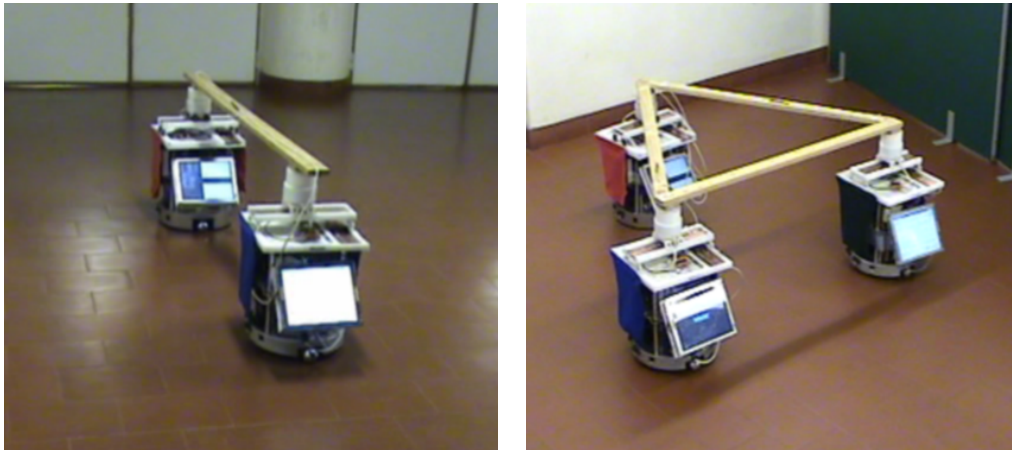
Neste capítulo são apresentadas algumas das características dos robôs que foram utilizados nesta dissertação. Em qualquer uma das equipas os robôs são semelhantes mas têm tarefas diferentes. É também abordada a comunicação entre os robôs bem como a sua cinemática.

---

### 3.1 Os robôs e as suas tarefas

Todos os robôs são idênticos, e formam equipas de dois ou três robôs conforme é ilustrado na figura 3.1. É nestes robôs móveis que se baseiam a implementação da arquitectura de controlo desenvolvida em ambiente de simulação por Soares (2007). Os três robôs foram construídos pela equipa de Robótica Autónoma e Sistemas Dinâmicos (sob a responsabilidade da Prof. Estela Bicho) do Departamento de Electrónica Industrial, da Escola de Engenharia da Universidade do Minho. O suporte de apoio ao objecto foi desenvolvido no âmbito desta dissertação, e é apresentado no capítulo 4.

O movimento de cada robô é controlado separadamente, daí se diz que a arquitectura de controlo utilizada é descentralizada, embora seja necessária a comunicação de algumas variáveis entre robôs.



(a) Equipa de dois robôs

(b) Equipa de três robôs

Figura 3.1: Equipas de robôs utilizados nesta dissertação. Na equipa de dois robôs, o robô a vermelho é o *Leader*, e o robô a azul é o *Helper*. Na equipa de três robôs, o robô a vermelho é o *Leader*, o robô a azul é o *Helper 1* e o robô a verde é o *Helper 2*.

Apesar de fisicamente todos os robôs serem idênticos, o robô *Leader* e os *Helpers* têm tarefas diferentes. O controlo e coordenação dos vários robôs, assenta no seguinte:

- o comportamento de cada robô é controlado individualmente;
- o robô *Leader* conhece a posição do alvo, e a sua tarefa consiste em navegar desde uma posição inicial até uma posição final designada por *tar*;
- o robô *Leader* comunica ao(s) robô(s) *Helper(s)* a sua velocidade e direcção em que “vê” o grupo;
- o robô *Leader* necessita de saber qual a dimensão do conjunto, objecto e restante equipa de robô(s). A dinâmica evitar colisões com obstáculos entra em linha de conta com a dimensão do conjunto para que não só o robô *Leader*, quando na presença de um obstáculo detectado, passe em segurança, mas também toda a equipa;
- cada robô *Helper* tem de manter em todos os instantes uma orientação e distância correctas em relação ao(s) seu(s) ponto(s) de referência;

- por defeito é imposto que cada robô *Helper* navegue na formação designada por *Forward/Backward*, no entanto se a presença de obstáculos não o permitir, o mesmo deverá navegar na formação *Turn* ou *Column*;
- o robô *Leader* comunica para o(s) *Helper(s)* a sua direcção de navegação e a sua velocidade linear;
- nas equipas constituídas por três robôs, os *Helpers* comunicam entre si as suas direcções de navegação, e o quão próximo os obstáculos são detectados;
- os robôs *Helpers* a partir da informação obtida através do suporte onde apoia o objecto a transportar, estão habilitados a medir a direcção na qual “vê(em)” o(s) seu(s) ponto(s) de referência, isto é, medir a direcção em que o(s) seu(s) ponto(s) de referência se encontra(m) relativamente a ele próprio.

## 3.2 Características

Cada robô é constituído por um conjunto de *hardware* que é descrito de seguida:

- 2 rodas motrizes diferenciais e os respectivos *drivers* de actuação;
- 1 computador;
- 1 placa de aquisição de dados de 16 bits de resolução;
- 7 sensores de distância, do tipo infravermelho;
- 6 baterias de 12 V (4 Ah);
- 2 conversores DC-DC, para adaptar a tensão das baterias para os níveis de tensão requeridos pelo *hardware*;
- 1 suporte para o objecto a transportar (para mais detalhe, ver o capítulo 4);

É de salientar que o computador utilizado no robô *Leader* é um *PC-104* a 133 MHz com 4 MB de RAM *onboard* e no caso dos robôs *Helpers* é utilizado um *Mini-itx* a 1.0 GHz com 256 MB de RAM e 40 GB de disco rígido.

No robô *Leader* o sistema operativo utilizado é o *MS-DOS 6.0* e o ambiente de programação é o *Turbo C*, enquanto que nos robôs *Helpers* o sistema operativo utilizado é o *Windows XP* e o ambiente de programação é o *MS Visual Studio*.

Os sensores de distância utilizados nos robôs são do tipo *GP2D12* da *Sharp*. Estes conseguem medir distâncias entre 10 e 80 cm. A curva característica da distância a que é detectado um objecto em função da tensão de saída do sensor, isto é,  $d[cm] = f(u[V])$ , referente ao tipo de sensor referido, está representada na figura 3.2.

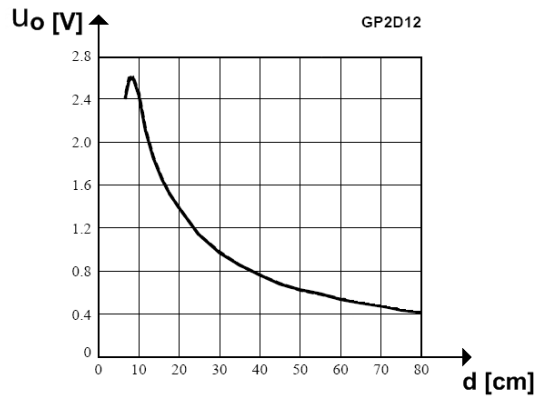


Figura 3.2: Curva característica do sensor *GP2D12* da *Sharp*.

De acordo com os testes realizados para a obtenção das características dos motores, registou-se uma velocidade máxima de 80 cm/s.

### 3.3 Comunicação

Independentemente do número de robôs que constituem a equipa, a comunicação entre o robô *Leader* e o(s) robô(s) *Helpers(s)* é unidireccional, isto é, o robô *Leader* comunica ao(s) robô(s) *Helper(s)* a sua velocidade linear e a direcção em que o grupo se encontra

relativamente a ele próprio. O robô *Leader* não tem em consideração nada do que se passa com a restante equipa, diz-se que este é “egoísta”. Esta característica tem como principal vantagem o fácil controlo do próprio robô *Leader*, e como principal desvantagem a tarefa tornar-se mais complicada para o(s) robô(s) *Helper(s)*.

Na equipa de dois robôs, o robô *Helper* utiliza a informação proveniente do robô *Leader* para determinar o valor desejado para a sua velocidade e ainda para determinar a formação desejada (para mais detalhe ver capítulo 6). Na equipa de três robôs a comunicação entre robôs *Helpers* é bidireccional. Estes comunicam entre si, o valor da sua direcção de navegação, o valor da sua função potencial e a magnitude da dinâmica referente ao comportamento *evitar colisões com obstáculos*. Esta informação é utilizada por cada um dos robôs *Helpers* para determinar o comportamento desejado (para mais detalhe ver capítulo 6).

### 3.4 Cinemática

A velocidade de translação,  $\vartheta$ , e a velocidade angular,  $\omega$ , neste tipo de plataformas são controladas através da velocidade de rotação das duas rodas dos robôs:

$$\omega_{wheel,left} = \frac{1}{R_{wheel}} \times \left( \vartheta + \frac{d_{wheel}}{2} \times \omega \right) \quad (3.1)$$

$$\omega_{wheel,right} = \frac{1}{R_{wheel}} \times \left( \vartheta - \frac{d_{wheel}}{2} \times \omega \right) \quad (3.2)$$

onde  $R_{wheel}$  representa o raio das rodas de cada um dos robôs e  $d_{wheel}$  representa a distância entre as rodas de cada um dos robôs.





## Capítulo 4

# Projecto do suporte

---

O suporte apresentado neste capítulo, teve como base um suporte anteriormente desenvolvido por Louro (2003), durante o seu estágio curricular, aquando da implementação de uma arquitectura de controlo para o transporte de um objecto de grandes dimensões por uma equipa de dois robôs móveis autónomos. Neste capítulo é apresentado o projecto do novo suporte, são descritas as suas peças constituintes, bem como os circuitos que o compõem.

---

### 4.1 Antigo suporte para o objecto

O suporte anteriormente desenvolvido por Louro (2003), para implementação de uma arquitectura de controlo para o transporte de um objecto por equipas de dois robôs, apenas tinha dois graus de liberdade: uma junta rotacional e uma junta prismática montada sobre a rotacional, como se pode observar na figura 4.1. O centro de rotação deste suporte era fixado coincidente com o eixo de rotação da plataforma robótica.

Uma solução em que o suporte para o objecto a transportar fosse baseada no suporte da figura 4.1 funcionava para equipas de dois robôs, mas no entanto, para equipas de maior número de robôs este tipo de suporte não se torna uma solução viável, porque apenas é

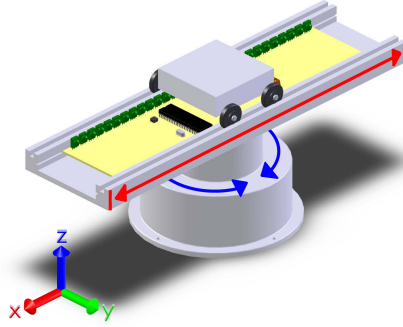


Figura 4.1: Antigo suporte para o transporte cooperativo por equipas de dois robôs.

capaz de sentir movimentos do grupo num eixo. Para equipas de dois robôs é suficiente a detecção de movimento apenas num só eixo, uma vez que só existe um agente externo a provocar movimento na junta prismática, estando esta sempre direccionada para o outro agente da equipa, devido à disposição das juntas. Uma vez que para equipas de maior número de robôs é necessário sentir o movimento em dois eixos, foi necessário desenvolver um novo suporte para apoiar o objecto a transportar.

## 4.2 Suporte para o objecto

O projecto do novo suporte consiste numa espécie de mesa XY, isto é, um tabuleiro com duas juntas prismáticas perpendiculares entre si, sendo ainda acoplada a estas uma junta rotacional, como se pode ver na figura 4.2.

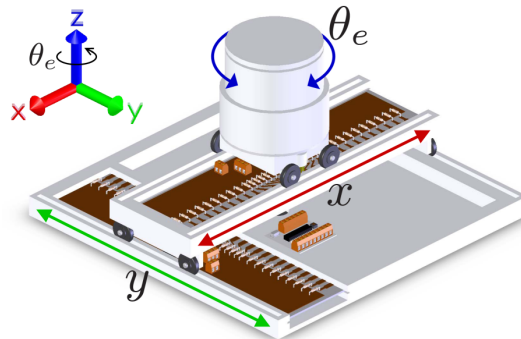


Figura 4.2: Suporte para apoiar o objecto a transportar para equipas de  $n$  robôs.

O suporte é montado na estrutura do robô de forma que fique centrado no centro deste (i.e., na origem dos eixos coordenados), e que o eixo  $Oy$  fique orientado com a direcção de navegação do robô. As juntas prismáticas indicam a posição em que o objecto a transportar se encontra em relação ao centro do próprio robô. A junta rotacional indica a direcção em que o objecto/grupo se desloca em relação às coordenadas em que o objecto se encontra no próprio robô.

Na figura 4.3 estão representados vários exemplos de posição e orientação de um objecto apoiado no suporte, em que a barra castanha representa o objecto a transportar. As coordenadas  $(x, y)$  representam o deslocamento relativamente ao centro do suporte e  $\theta_e$  representa a rotação em torno do eixo  $Oz$  medido relativamente ao eixo  $Ox$ .

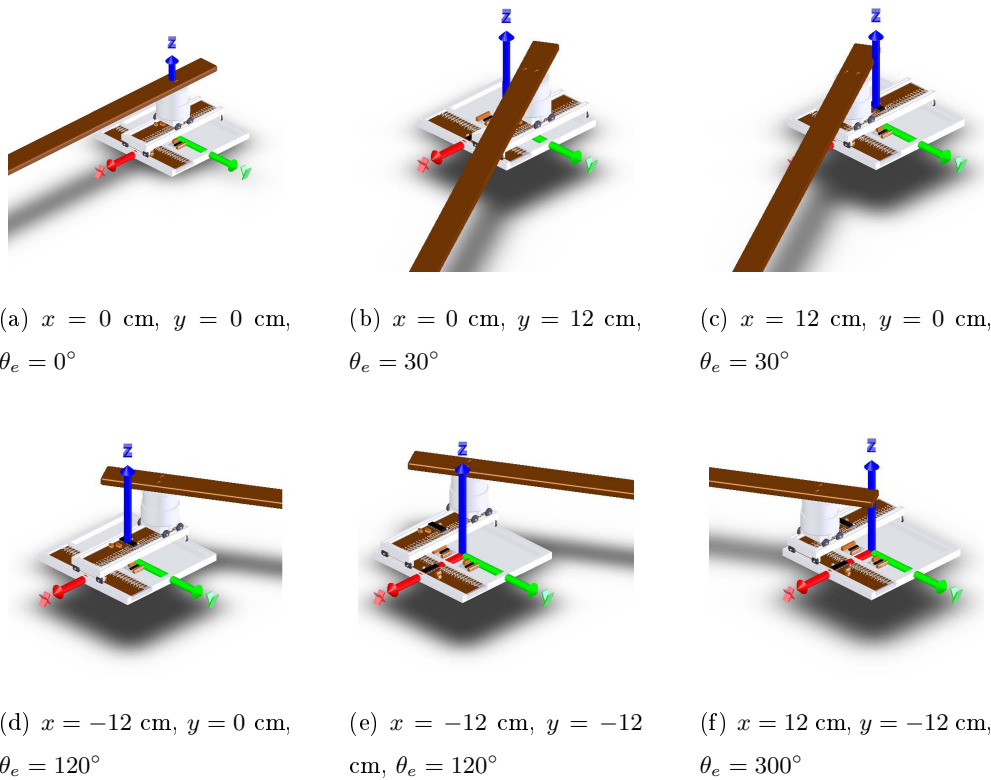


Figura 4.3: Exemplos de posições das juntas do suporte, em que  $\{x, y\}$  representam as coordenadas em que o suporte do objecto se encontra em relação à origem dos eixos coordenados e  $\theta_e$  representa o ângulo de rotação em torno do eixo  $Oz$ , medido em relação ao eixo  $Ox$ .

### 4.2.1 Juntas prismáticas

As juntas prismáticas são, cada uma, compostas por dois circuitos, um responsável pela leitura da posição e o outro responsável pela conversão série-paralelo da mesma leitura.

Uma vez que os controladores de interface programável (PICs) utilizados para efectuar a leitura da posição não tinham linhas de entradas e/ou saídas suficientes para enviar a informação directamente para a placa de aquisição de dados, foi necessário recorrer à implementação de um circuito de conversão série-paralelo. A comunicação entre estes dois circuitos é efectuada via porta série.

#### 4.2.1.1 Leitura de uma junta

Na figura 4.4 está representado o modelo CAD da PCB de uma junta prismática onde se podem ver as ligações externas à PCB. A porta série é o conector laranja onde estão os fios roxo e verde, sendo eles a linha de *Transmit* e de *Receive*, respectivamente, da porta série do PIC. A alimentação é no conector onde estão as ligações a vermelho e pela preto, sendo elas  $+5V$  e *Gnd*, respectivamente. É de notar que o rolamento (parte metálica que desliza sobre os contactos C01 a C29) está alimentado a  $+5V$ . O circuito de interface programável (PIC) existente tem como função efectuar ciclicamente a leitura da posição em que o rolamento se encontra, armazená-la, e quando o PIC do circuito que será apresentado de seguida efectuar um pedido, este responde com o valor armazenado, que é a posição mais recente.

Na figura 4.5 está representado o esquema eléctrico da PCB de uma junta prismática. Os contactos C01 a C29 representam os possíveis contactos entre as pistas da PCB e a parte metálica (um rolamento) alimentada do carrinho, constituindo assim estes contactos juntamente com uma peça metálica alimentada a  $+5V$  uma régua instrumentada. Nessa figura, *S1* e *S2* representam a saída *Transmit* e a entrada *Receive*, respectivamente, da porta série do *PIC16F874*.

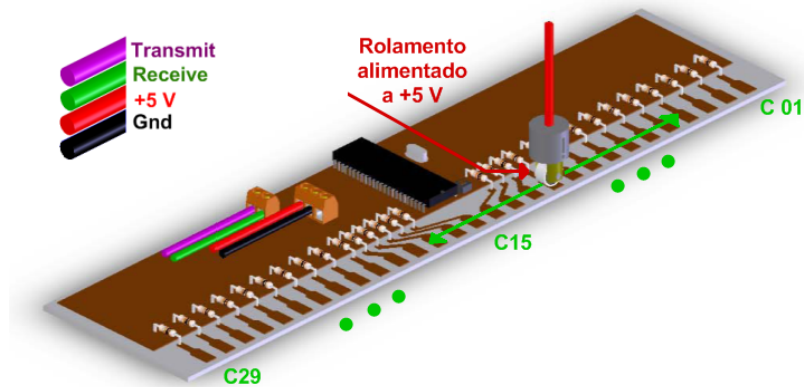


Figura 4.4: Modelo CAD da PCB de uma junta prismática e respectivas entradas/saídas.

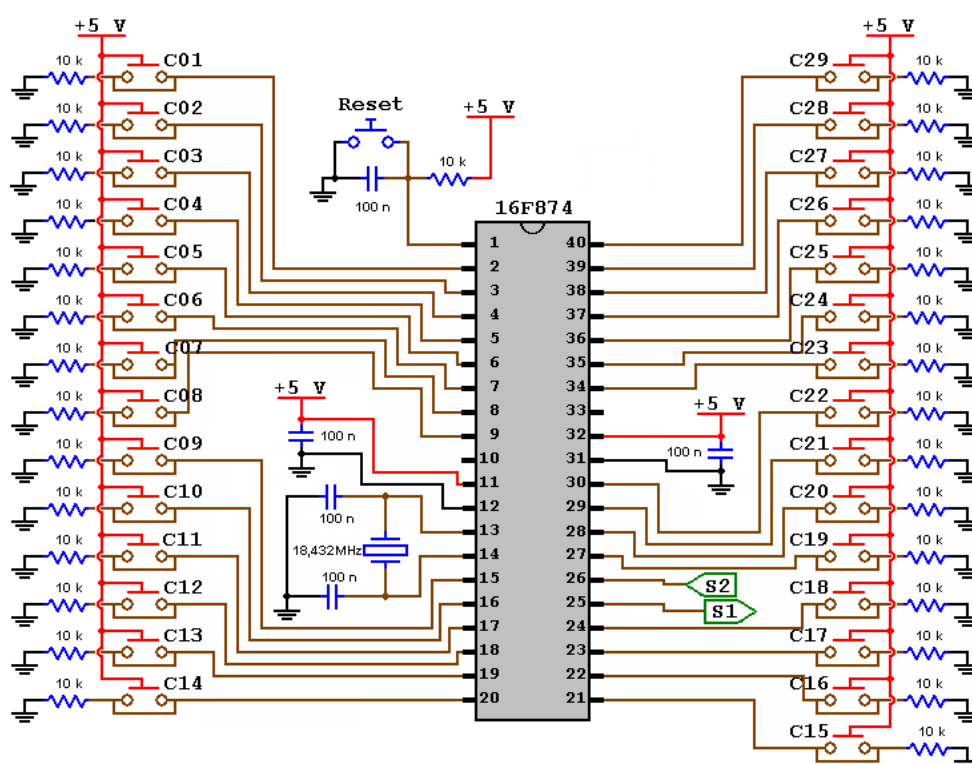


Figura 4.5: Esquema eléctrico para efectuar a leitura de uma junta prismática.

## Programação do PIC 16F874

Na figura 4.6 está representado o fluxograma correspondente à programação do PIC 16F874 que é responsável pela leitura da posição de uma junta prismática.

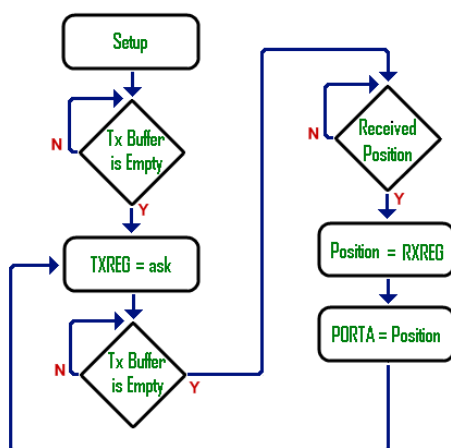


Figura 4.6: Fluxograma do software que corre no *PIC16F874*.

Inicialmente é feito o *setup* ao circuito de interface programável (PIC), isto é, a configuração dos portos, da comunicação e das interrupções. De seguida o PIC fica em *loop* a efectuar a leitura da posição da junta, guardando este valor numa posição de memória. Quando o circuito de conversão série-paralelo (descrito mais à frente) faz um pedido, via porta série, este responde-lhe também via porta série com o valor mais recente guardado referente à última posição lida, regressando de seguida ao referido *loop* de leitura da posição.

### 4.2.1.2 Conversão série-paralelo da leitura da junta

Devido à falta de portas séries livres nos robôs, foi necessário efectuar a leitura das juntas prismáticas através da placa de aquisição de dados. Para tal, foi necessário proceder à conversão série - paralelo uma vez que as portas digitais livres do circuito da figura 4.5 não eram suficientes. Para tal conversão foi projectado um circuito, figura 4.7, para efectuar a conversão.

Na figura 4.7, *S1* e *S2* representam a entrada *Receive* e a saída *Transmit*, respectivamente, da porta série do PIC16F876. *Bit 4*, *Bit 3*, *Bit 2*, *Bit 1* e *Bit 0* representam as saídas que compõem a palavra de 5 bits da informação que será recolhida através da placa de aquisição de dados, sendo *Bit 4* e *Bit 0* o *MSB* (*Most Significant Bit*) e *LSB* (*Least Significant Bit*), respectivamente.

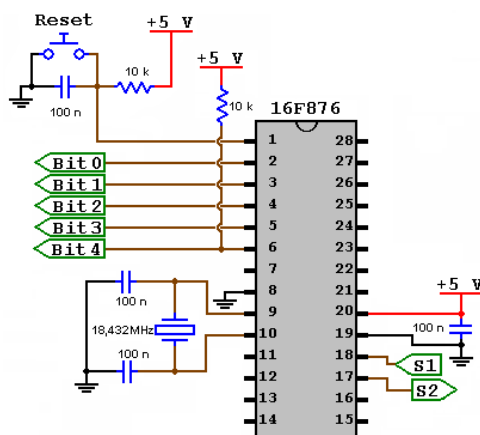


Figura 4.7: Esquema eléctrico para efectuar a conversão Série - Paralelo da informação da junta prismática.

Na figura 4.8 está representada uma PCB cuja função é a conversão série-paralelo da informação recebida da PCB que efectua a leitura da posição de uma junta prismática. Nesta figura estão também representadas as ligações a efectuar na PCB de conversão. A porta série é o conector laranja que contém os fios roxo e verde, sendo eles a linha de *Receive* e de *Transmit*, respectivamente, da porta série do PIC. A alimentação é no conector onde estão as ligações a vermelho e pela preto, sendo elas +5V e *Gnd*, respectivamente. As saídas azul celeste, verde lima, azul, verde e castanho, representam o *Bit 4*, *Bit 3*, *Bit 2*, *Bit 1* e *Bit 0*, respectivamente, da palavra que constitui o valor lido de uma junta prismática.

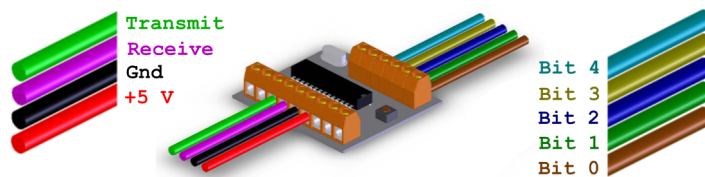


Figura 4.8: Modelo CAD do PCB responsável pela conversão série-paralelo do valor da junta prismática.

### Programação do PIC 16F876

Na figura 4.9 está representado o fluxograma correspondente à programação do PIC 16F876 que é responsável pela conversão série-paralelo do valor lido por uma junta prismática. Inicialmente é feito o *setup* ao circuito de interface programável (PIC), isto é, a configuração dos portos e da comunicação. De seguida o PIC fica em ciclo a efectuar os pedidos da posição actual da junta, via porta série, e recebe também o valor da posição da junta, via porta série, disponibilizando esse valor nas saídas *Bit 4*, *Bit 3*, *Bit 2*, *Bit 1* e *Bit 0*, estando assim deste modo sempre disponível na placa de aquisição de dados o valor da posição da junta actualizado.

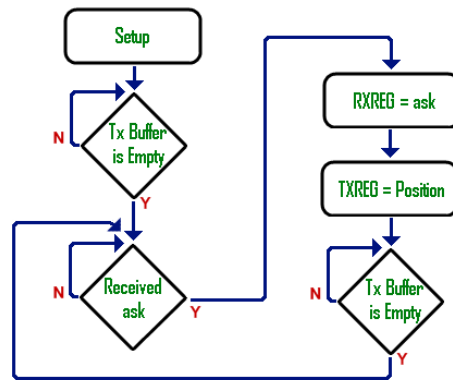


Figura 4.9: Fluxograma do software que corre no *PIC16F876*.

#### 4.2.2 Junta rotacional

A utilização de uma junta rotacional, figura 4.10, permite a cada robô saber em que direcção o objecto a transportar, por conseguinte o(s) elemento(s) da equipa, se encontra(m) em relação a si próprio.

Por forma a poder saber-se em que posição se encontra a junta rotacional houve necessidade de colocar internamente um *encoder*.

O *encoder* utilizado na junta rotacional foi o *Absolute Encoder A2* da *US Digital*. Para efectuar uma leitura da posição da junta rotacional é necessário enviar o *byte 0x1F*, e o



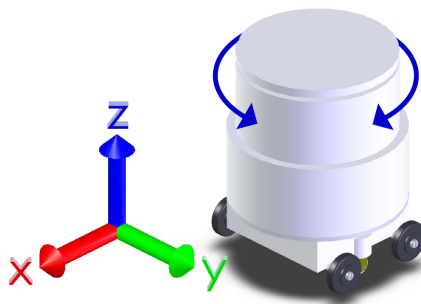


Figura 4.10: Junta rotacional.

*encoder* responde com três *bytes*, sendo o primeiro *byte* o *acknowledge*, que é o mesmo que o *byte* do pedido da posição, o segundo e o terceiro *bytes* são os de informação. A informação recebida são valores entre 0000 e 3599. O segundo *byte* recebido corresponde aos dois *bits* mais significativos (MSB), e o terceiro *byte* recebido corresponde aos dois *bits* menos significativos (LSB). Suponha-se que se pretende efectuar uma leitura da posição da junta, para tal é necessário o seguinte: enviar o pedido de leitura, isto é enviar o *byte*  $0x1F$ , e efectuar a recepção do *acknowledge*, caso este seja igual ao *byte* do pedido de leitura, então efectua-se a leitura dos dois *bytes* de informação, caso contrário não. Se o segundo *byte* for o número *mc* e o terceiro *byte* for o número *du*, então o número lido é o  $mc \times 100 + du = mcdu$ , em que *m* é o algarismo dos milhares, *c* o das centenas, *d* o das dezenas e *u* o das unidades. Uma vez que o valor recebido está sempre compreendido entre 0 e 3600, para obter o ângulo em que a junta rotacional se encontra é necessário dividir o valor *mcdu* por 10, para obter um ângulo entre  $[0^\circ; 360^\circ]$ .

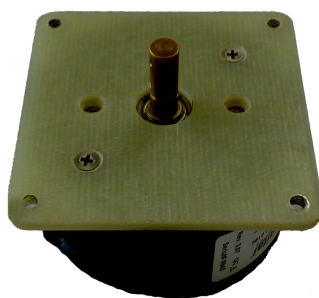


Figura 4.11: *Encoder* absoluto utilizado na junta rotacional, A2 da *US Digital*.

A entrada/saída deste *encoder* é do tipo *SEI bus* por conseguinte houve também a necessidade de adquirir um conversor (neste caso foi adquirido o conversor *SEI bus*

- *RS232*) por forma a poder fazer a aquisição através do PC. Este conversor tem a referência *AD2B* da *US Digital*, figura 4.12. O conversor é alimentado a +12V.



Figura 4.12: Conversor *SEI bus* para *RS232* da *US Digital*.

Na figura 4.13 pode-se ver o esquema de ligações do *encoder* com o conversor e a respectiva alimentação.

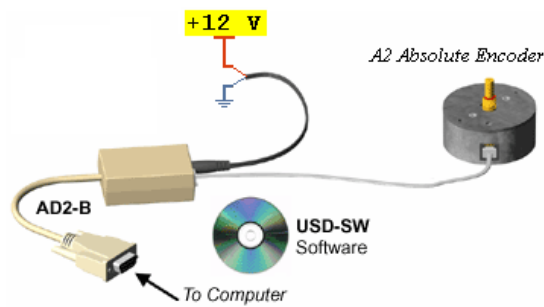


Figura 4.13: Esquema de ligações do *encoder* da junta rotacional.

## Capítulo 5

# Arquitectura de controlo do robô

## *Leader*

---

A arquitectura de controlo aqui usada para a geração do movimento do *Leader* foi previamente desenvolvida e aplicada por Bicho (2000). Esta arquitectura é baseada na abordagem dinâmica para a robótica, Schöner *et al.* (1995).

---

### 5.1 Dinâmica de atractores para a direcção de navegação

O movimento do robô é gerado através da atribuição contínua no tempo de valores às variáveis controladas: direcção de navegação e velocidade linear/translação. O percurso temporal de cada uma destas variáveis é obtido através de soluções constantes de sistemas dinâmicos.

Os estados assintoticamente estáveis, isto é, as soluções atractoras, dominam por princípio de *design*. A dinâmica comportamental da direcção de navegação,  $\phi_L(t)$ , e da velocidade,  $\vartheta_L(t)$  são equações diferenciais:

$$\frac{d\phi_L(t)}{dt} = f(\phi_L(t), \text{parâmetros}) \quad (5.1)$$

$$\frac{d\vartheta_L(t)}{dt} = g(\vartheta_L(t), \text{parâmetros}) \quad (5.2)$$

Como se pode ver na figura 5.1, a direcção especificada por  $\psi_{tar}$ , direcção em que está o alvo a atingir, é uma direcção desejada, enquanto que a direcção especificada por  $\psi_{obs}$ , direcção em que se encontra um obstáculo, é uma direcção a evitar.

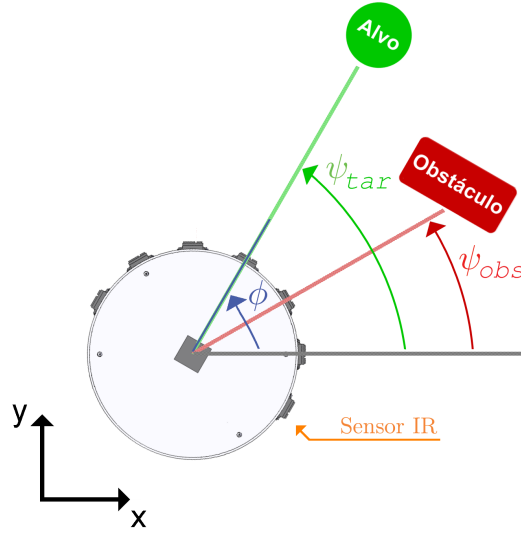


Figura 5.1: Valores das direcções que entram no cálculo da dinâmica comportamental.

### 5.1.1 Comportamento *dirigir-se para o alvo*

Para calcular a direcção em que o robô *Leader* “vê” o alvo utiliza-se a equação trigonométrica

$$\psi_{tar} = \arctan_2 \left( \frac{y_{tar} - y_{robot}}{x_{tar} - x_{robot}} \right) \quad (5.3)$$

onde  $(x_{tar}, y_{tar})$  especificam as coordenadas do alvo, assumindo que estas são conhecidas, relativamente a um sistema de eixos coordenados externo arbitrário, mas fixo, e  $(x_{robot}, y_{robot})$  as coordenadas do robô, relativamente ao mesmo sistema eixos. As coordenadas do robô são obtidas através do mecanismo de hodometria:

$$\frac{dx_L(t)}{dt} = \vartheta_L(t) \times \cos(\phi_L) \quad , \quad \phi_L \in [0; 2\pi] \text{ rad} \quad (5.4)$$

$$\frac{dy_L(t)}{dt} = \vartheta_L(t) \times \sin(\phi_L) \quad , \quad \phi_L \in [0; 2\pi] \text{ rad} \quad (5.5)$$

que discretizando pelo método de *Euler-progressivo*, obtém-se

$$x_L(t_k) = x_L(t_{k-1}) + \vartheta_L(t_{k-1}) \times \cos(\phi_L(t_{k-1})) \times dt \quad , \quad \forall k \in N^+ \quad (5.6)$$

$$y_L(t_k) = y_L(t_{k-1}) + \vartheta_L(t_{k-1}) \times \sin(\phi_L(t_{k-1})) \times dt \quad , \quad \forall k \in N^+ \quad (5.7)$$

onde  $t_k$  representa o ciclo computacional actual, e  $t_{k-1}$  representa o ciclo computacional anterior,  $\vartheta_L$  a velocidade linear,  $\phi_L$  a direcção de navegação do robô *Leader* e  $dt$  representa o tempo que o ciclo computacional demorou.

A orientação na direcção do alvo é especificada através da implementação de um sistema dinâmico que cria um atractor na direcção  $\psi_{tar}$  (como se pode ver na figura 5.2).

$$\frac{d\phi_L(t)}{dt} = f_{tar}(\phi_L(t)) = -\lambda_{tar} \times \sin(\phi_L(t) - \psi_{tar}) \quad , \quad \lambda_{tar} > 0 \quad (5.8)$$

em que  $\lambda_{tar}$  representa a magnitude de atracção (taxa de relaxação local),  $\phi_L$  a actual direcção de navegação do robô *Leader* e  $\psi_{tar}$  a direcção segundo a qual o robô *Leader* “vê” a posição alvo.

No caso da existência apenas do comportamento *dirigir-se para o alvo* a velocidade de rotação, é dada pelo valor de  $-\lambda_{tar} \times \sin(\phi_L(t_{k-1}) - \psi_{tar})$ .

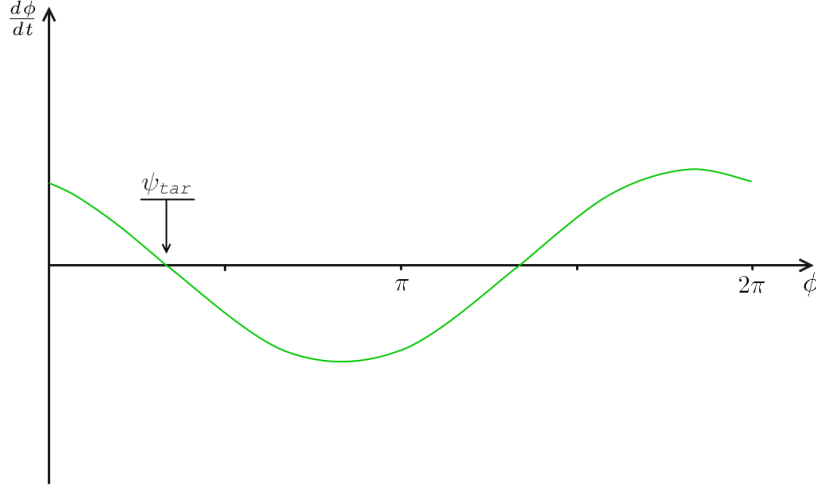


Figura 5.2: Atractor na direcção do alvo da figura 5.1,  $\psi_{tar}$ . A direcção  $\phi_L = \phi_{tar}$  é um ponto de equilíbrio atractor, isto é,  $\frac{d\phi}{dt} = 0$  com declive negativo, com uma magnitude de atracção  $\lambda_{tar}$ . Como o movimento na direcção do alvo é desejado, qualquer que seja a orientação inicial do robô, o alcance do atractor, isto é, o efeito de atracção provocado pelo ponto de equilíbrio atractor é o círculo trigonométrico completo. Como consequência, existe um ponto de equilíbrio repulsor na direcção oposta ao ponto atractor, isto é, na direcção  $\psi_{tar} + \pi$ .

### 5.1.2 Comportamento *evitar colisões com obstáculos*

Quando o robô se encontra em movimento, este deve evitar colisões com possíveis obstáculos que lhe surjam no seu trajecto. As direcções em que os obstáculos são detectados devem ser evitadas. Para tal utiliza-se um sistema dinâmico que repele o robô dessas direcções. Os sensores de distância estão montados no robô com um ângulo  $\theta_i$ , em relação à direcção frontal do robô.

$$\theta_i = (-4 + i) \times \Delta\theta, \quad i = [1; 7] \forall i \in N \quad (5.9)$$

onde  $\Delta\theta$  representa o espaçamento angular entre cada par consecutivo de sensores de distância, sendo nestes robôs de  $\frac{\pi}{6} \text{ rad}$ , isto é,  $30^\circ$ .

Dispondo sete sensores de distância, como indicado na figura 5.3, cuja gama angular é de  $\frac{\pi}{6}$  consegue-se uma cobrir toda a frente do robô com o número mínimo de sensores.

Caso um determinado sensor de distância detecte uma obstrução, esse contribui com

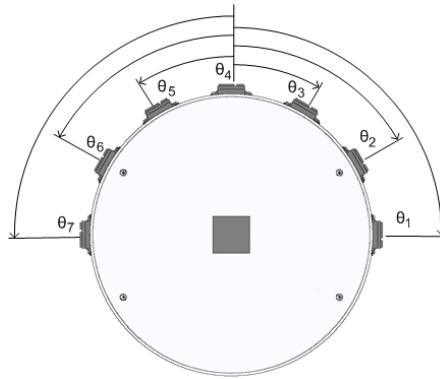


Figura 5.3: Disposição dos sensores, segundo um ângulo  $\theta_i$ , em relação à direcção frontal do robô.

uma força repulsiva centrada na direcção segundo a qual o sensor aponta. Para melhor exemplificar isto, tome-se como exemplo a figura 5.1, a qual resulta uma força repulsiva como a representada na figura 5.4.

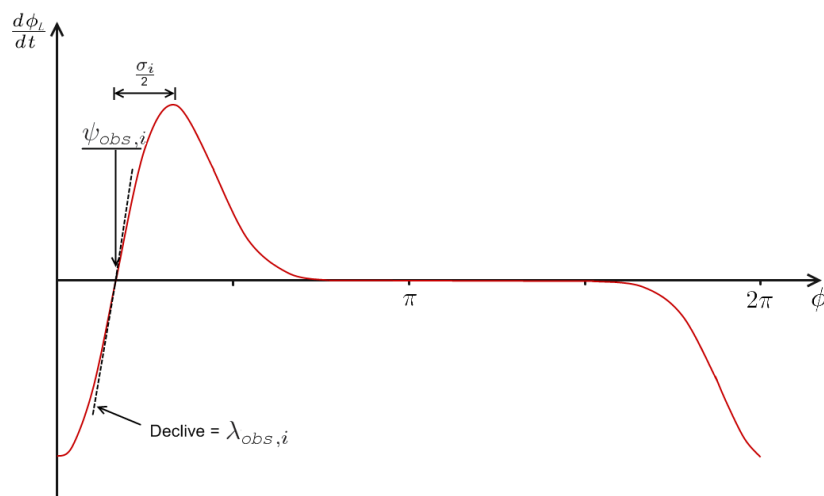


Figura 5.4: A contribuição para a dinâmica da direcção de navegação correspondente à tarefa *evitar colisões com obstáculos* resulta numa força com um ponto de equilíbrio repulsor, isto é,  $\frac{d\phi_i}{dt} = 0$  com declive positivo, na direcção  $\psi_{obs,i}$ , na qual é detectada uma obstrução, representando  $i$  cada sensor de distância.

Ajustando adequadamente o valor do parâmetro  $\lambda_{obs,i}$ , faz-se com que apenas obstruções próximas repelem fortemente o robô da direcção  $\psi_{obs,i}$ , i.e., da direcção em que a obstrução é detectada. O alcance destas forças repulsivas,  $\sigma_i$ , é limitado pelo alcance dos sensores e no princípio de que o robô passe pelo obstáculo sem ter qualquer contacto com este.

A dinâmica resultante da tarefa evitar colisões com obstáculos resulta no sistema dinâmico

$$\frac{d\phi_L(t)}{dt} = \sum_{i=1}^7 f_{obs,i}(\phi_L) \quad (5.10)$$

onde,

$$f_{obs,i}(\phi_L) = \lambda_{obs,i} \times (\phi_L - \psi_{obs,i}) \times e^{\left[-\frac{(\phi_L - \psi_{obs,i})^2}{2 \times \sigma_i^2}\right]}, \quad i = [1; 7] \forall i \in N \quad (5.11)$$

A direcção  $\psi_{obs,i}$  representa a direcção no mundo na qual o sensor de distância  $i$  aponta. Como a direcção de navegação do robô,  $\phi_L$ , é definida relativamente ao mesmo eixo de referência, então a diferença  $\phi_L - \psi_{obs,i} = -\theta_i$  é na realidade uma constante, sendo  $\theta_i$  dado pela equação 5.9, em que  $\Delta\theta$  representa o espaçamento angular dos sensores. Uma vez que o cálculo do valor da direcção de navegação é feito pelo método de hodometria, este acumula muito erro ao longo do tempo, então tenta-se evitar a utilização do valor de  $\phi_L$ . Assim sendo, a calibração da direcção de navegação do robô, no que respeita à tarefa de *evitar colisões com obstáculos*, é irrelevante.

Substituindo a diferença  $\phi_L - \psi_{obs,i}$  na equação 5.11 por  $-\theta_i$  tem-se:

$$f_{obs,i}(\phi_L) = -\lambda_{obs,i} \times \theta_i \times e^{\left[-\frac{\theta_i^2}{2 \times \sigma_i^2}\right]}, \quad i = [1; 7] \forall i \in N \quad (5.12)$$

A magnitude de repulsão,  $\lambda_{obs,i}$ , de cada uma das contribuições é uma função que decresce com o aumento da distância aos obstáculos a qual depende de dois parâmetros: da magnitude máxima de repulsão,  $\beta_1 > 0$  e da taxa de decaimento da magnitude de repulsão,  $\beta_2 > 0$  em função do aumento da distância ao obstáculos,  $d_i$ .

$$\lambda_{obs,i} = \beta_1 \times e^{-\frac{d_i}{\beta_2}}, \quad i = [1; 7] \forall i \in N \quad (5.13)$$



A gama angular sobre a qual uma força repulsiva exerce o seu efeito é controlada por  $\sigma_i$  que depende do ângulo de sensibilidade do sensor de distância,  $\Delta\theta$ , do valor de *Radius*, e também da distância medida,  $d_i$ .

$$\sigma_i = \arctan \left[ \tan \left( \frac{\Delta\theta}{2} \right) + \frac{Radius}{Radius + d_i} \right], \quad i = [1; 7] \forall i \in N \quad (5.14)$$

A gama angular sobre a qual uma força repulsiva actua diminui com o aumento da distância,  $d_i$ , porque o ângulo subentendido pelo próprio robô e restante equipa diminui.

*Radius*, entra em linha de conta com o raio de uma circunferência que abranja toda a equipa, pois aos robôs *Helper(s)* apenas lhe cabe a tarefa de ajudar o robô *Leader* a transportar o objecto. Assim, através da gama angular sobre a qual a força repulsiva, do robô *Leader*, exerce o seu efeito, consegue-se evitar que este entra em passagens estreitas em que a equipa não tem condições para passar pela mesma.

### 5.1.3 Integração dos dois comportamentos, *dirigir-se para o alvo e evitar colisões com obstáculos*

A contribuição do alvo e a contribuição resultante dos obstáculos actuam simultaneamente (figura 5.5). A direcção de navegação do robô *Leader*,  $\phi_L$ , resulta da soma destas duas contribuições

$$\frac{d\phi_L(t)}{dt} = f_{tar}(\phi_L) + \sum_{i=1}^7 f_{obs,i}(\phi_L) + f_{stoch} \quad (5.15)$$

Usando o método de *Euler progressivo* para discretizar a equação da dinâmica resultante tem-se que:

$$\phi_L(t_k) = \phi_L(t_{k-1}) + \left( f_{tar}(\phi_L(t_{k-1})) + \sum_{i=1}^7 -\lambda_{obs,i} \times \theta_i \times e^{\left[ -\frac{\theta_i^2}{2 \times \sigma_i^2} \right]} + f_{stoch} \right) \times dt \quad (5.16)$$

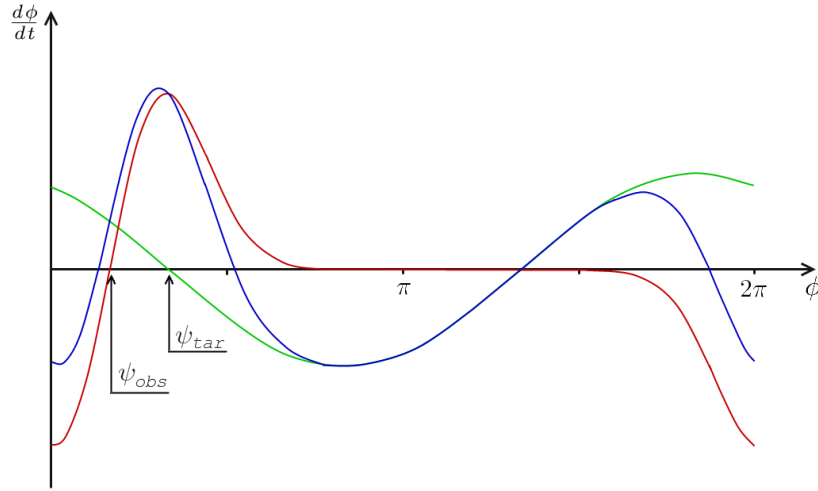


Figura 5.5: Da integração do comportamento *dirigir-se para o alvo*, função a verde, com o comportamento *evitar colisões com obstáculos*, função a vermelho, resulta a dinâmica comportamental final do robô *Leader*, função a azul. Esta última resulta da soma ponto a ponto de ambas as outras funções.

À soma da contribuição do alvo e da resultante dos obstáculos é ainda adicionado uma força estocástica,  $f_{stoch}$ , porque caso o sistema ficasse preso num ponto de equilíbrio repulsor com a adição desta força evita que tal aconteça. Em termos prático não é necessário a utilização do membro estocástico, pois a própria aquisição sensorial já tem ruído estocástico que assume esta função. Esta força estocástica pode ser utilizada para introduzir ruído no sistema de controlo por forma a testar a sua robustez.

Por princípio de *design*, os parâmetros são sintonizados por forma a que a direcção de navegação,  $\phi_L$ , esteja sempre perto de um atractor da dinâmica resultante.

## 5.2 Controlo da velocidade

Até esta secção, apenas foi referido o controlo da direcção de navegação do robô *Leader*,  $\phi_L$ , no entanto, para que o robô se movimente, num plano horizontal, é necessário controlar a velocidade de translação do mesmo. À medida que o robô se move a informação sensorial varia, assim os atractores e os repulsores também variam. O mesmo acontece se os obstáculos e/ou o alvo se movimentarem no ambiente em que o robô está. Para que a direcção de navegação do robô, por *design* esteja muito próxima, ou até mesmo

num atractor da dinâmica resultante, há necessidade de limitar a taxa de variação dos pontos atractores para permitir que a direcção de navegação do robô consiga perseguir o atractor para se manter sempre num estado estável. Isto é conseguido através de um sistema dinâmico que controla a velocidade,  $\vartheta_L$ ,

$$\frac{d\vartheta_L(t)}{dt} = g(\vartheta_L) = -\lambda_v \times (\vartheta_L - \vartheta_{L,desejada}) \quad (5.17)$$

onde  $g(\vartheta_L)$ , representa um campo vectorial que erige um atractor na velocidade desejada,  $\vartheta_{L,desejada}$ , com uma magnitude  $\lambda_v$ .

Quanto à velocidade desejada para o robô *Leader*,  $\vartheta_{L,desejada}$ , esta depende se a direcção de navegação do robô *Leader* está numa zona repulsiva criada pela presença de obstáculos. A velocidade desejada para o robô *Leader* é dada por:

$$\vartheta_{L,desejada} = \begin{cases} \frac{d_{min,L,obs}}{T_{2c}} & , \text{ se } U_{obs}(\phi_L) > 0 \\ K_v & , \text{ caso contrário} \end{cases} \quad (5.18)$$

onde  $d_{min,L,obs}$  representa a menor distância entre o robô e os obstáculos,  $T_{2c}$  define o tempo para o contacto com o obstáculo detectado, e  $U_{obs}(\phi_L)$  é a função potencial da dinâmica *evitar colisões com obstáculos* dada pela equação 5.19.  $K_v$  é uma constante que corresponde à velocidade do robô *Leader* quando este não está na presença de obstáculos

$$U_{obs}(\phi_L) = \sum_{i=1}^7 \left( \lambda_{obs,i} \times \sigma_i^2 \times e^{\left[ -\frac{(\phi_L - \psi_{obs,i})^2}{2 \times \sigma_i^2} \right]} - \frac{\lambda_{obs,i} \times \sigma_i^2}{\sqrt{e}} \right) \quad (5.19)$$

onde  $\psi_{obs,i}$  é a direcção na qual o sensor de distância  $i$  do robô *Leader* aponta,  $\lambda_{obs,i}$  é a magnitude de repulsão e  $\sigma_i$  é a gama angular de repulsão.  $\lambda_{obs,i}$  e  $\sigma_i$  são dados pela equações 5.13 e 5.14, respectivamente.

Valores positivos da equação 5.19 indicam que a direcção de navegação do robô está

dentro de uma região suficientemente forte, criada pela presença de obstáculos. Contrariamente, valores não positivos indicam que a mesma direcção está fora de uma zona de repulsão considerada fraca ou mesmo nula.

Discretizando a equação 5.17 pelo método de *Euler progressivo*, tem-se

$$\vartheta_L(t_k) = \vartheta_L(t_{k-1}) - \lambda_v \times [\vartheta_L(t_{k-1}) - \vartheta_{L,desired}(t_k)] \times dt, \quad \forall k \in N^+ \quad (5.20)$$

### 5.3 Hierarquia das taxas de relaxação

Para assegurar que a direcção de navegação relaxa para as soluções atractoras à medida que estas variam, devido à variação da informação sensorial e tendo como princípio que o comportamento *evitar colisões com obstáculos* tem prioridade sobre o comportamento *dirigir-se para o alvo*, é necessário respeitar a hierarquia das taxas de relaxação:

$$\lambda_{tar} < \lambda_{obs} < \lambda_v \quad (5.21)$$

Por forma a respeitar esta hierarquia e tendo em conta que o tempo de um ciclo computacional ronda os 50 *ms*, foram atribuídos aos parâmetros da dinâmica os seguintes valores (para mais detalhes ver Bicho (2000)):

$$\lambda_{tar} = \frac{1}{10 \times dt} = 2;$$

$$T_{2c} = \frac{1}{2} \times dt = 25 \times 10^{-3};$$

$$\beta_1 = \frac{1}{T_{2c}} = \frac{1}{25 \times 10^{-3}} = 40;$$

$$\beta_2 = 25;$$

$$\lambda_v = 30;$$

## Capítulo 6

# Arquitectura de controlo para o(s) robô(s) *Helper(s)*

---

A arquitectura de controlo usada para a coordenação do movimento dos robôs, *Leader* e *Helper(s)*, foi desenvolvida e testada em ambiente de simulação por Soares (2007), no âmbito do seu doutoramento. À semelhança do robô *Leader*, também para o(s) robô(s) *Helper(s)* foram escolhidas como variáveis controladas a direcção de navegação,  $\phi_{H_{any}}$ , e a velocidade linear,  $v_{H_{any}}$ . Ao longo deste capítulo, quando for utilizado o índice *any* quer-se referir a qualquer robô *Helper*, isto é a variável/parâmetro refere-se tanto ao robô *Helper* da equipa de dois robôs como aos robôs *Helper 1* e *Helper 2* da equipa de três robôs.

---

Para as equipas de robôs móveis autónomos foram implementados três tipos de comportamentos, sendo eles o *Forward/Backward*, o *Turn* e o *Column*. Por defeito, isto é, sempre que seja possível, o(s) robô(s) *Helper(s)* deve(m) navegar no comportamento *Forward/Backward*, e quando não seja possível, este(s) deve(m) navegar no comportamento *Turn* ou *Column*, dependendo de como são sentidos os obstáculos pelos próprios robô(s) *Helper(s)*, de forma a evitar(em) os obstáculos.

Para controlar o movimento do(s) robô(s) *Helper(s)* usou-se a direcção de navegação do respectivo robô *Helper*,  $\phi_{H_{any}}$  e a respectiva velocidade linear,  $\vartheta_{H_{any}}$ .

A dinâmica para a direcção de navegação de cada robô *Helper* é dada pela equação:

$$\frac{d\phi_{H_{any}}(t)}{dt} = -\lambda_{H_{any}} \times \sin(\phi_{H_{any}} - \psi_{desired,H_{any}}), \quad \lambda_{H_{any}} > 0 \quad (6.1)$$

a qual erige um atractor na direcção  $\psi_{desired,H_{any}}$ , e um repulsor na direcção oposta, com uma magnitude de atracção definida por  $\lambda_{H_{any}}$ .

A dinâmica para o controlo da velocidade linear coloca um atractor na velocidade desejada,  $V_{desired,H_{any}}$ , com uma magnitude de atracção definida por  $\lambda_v$ .

$$\frac{d\vartheta_{H_{any}}(t)}{dt} = -\lambda_v \times (\vartheta_{H_{any}} - V_{desired,H_{any}}), \quad \lambda_v > 0 \quad (6.2)$$

## 6.1 Dinâmica de atractores para a direcção de navegação

O atractor  $\psi_{desired,H_{any}}$ , do sistema dinâmico definido pela equação 6.1, é dado por:

$$\psi_{desired,H_{any}} = \begin{bmatrix} \gamma_{H_{any},f/b} & \gamma_{H_{any},t} & \gamma_{H_{any},c} \end{bmatrix} \times \begin{bmatrix} \psi_{desired,H_{any},f/b} \\ \psi_{desired,H_{any},t} \\ \psi_{desired,H_{any},c} \end{bmatrix} \quad (6.3)$$

onde  $\psi_{desired,H_{any},f/b}$ ,  $\psi_{desired,H_{any},t}$  e  $\psi_{desired,H_{any},c}$  são as direcções desejadas nas quais os atractores são erigidos para os comportamentos *Forward/Backward*, *Turn* e *Column*, respectivamente.  $\gamma_{H_{any},f/b}$ ,  $\gamma_{H_{any},t}$  e  $\gamma_{H_{any},c}$  são variáveis de activação mutuamente exclusivas, destes mesmos comportamentos, que dependendo da informação sensorial e/ou comunicada, determina qual o comportamento que deve estar activo.

### 6.1.1 Para a equipa de dois robôs

O valor desejado para o ângulo segundo o qual o atractor da dinâmica para o robô *Helper* deve ser erigido é dado pela equação:

$$\psi_{desired,H,behaviour} = \psi_{H,L} + \Delta\psi_{H,L,behaviour} + R_{H,L,behaviour} \times \Delta_H \quad (6.4)$$

onde  $\psi_{H,L}$  é a direcção segundo a qual o robô *Helper* “vê” o robô *Leader* (ver figura 6.1 e equação 6.5), este ângulo é medido de centro a centro de cada robô,  $\Delta\psi_{H,L,behaviour}$  é um ângulo característico do comportamento *behaviour*. *behaviour* pode ser *f/b*, *t* ou *c*, representando os comportamentos *Forward/Backward*, *Turn* ou *Column*, respectivamente.  $\Delta_H$  representa o ângulo que permite compensar o deslocamento do suporte do objecto (que é dado pela equação 6.6).  $R_{H,L,behaviour}$  também é definido para cada comportamento, como se pode ver nas subsubsecções referentes aos mesmos.  $R_{H,L,behaviour}$  é um parâmetro que indica ao sistema de controlo se o robô está a uma distância maior ou menor que a distância desejada ao robô *Leader*, fazendo com que  $\Delta_H$  (ver equação 6.6) seja somado ou subtraído, permitindo ao robô *Helper* afastar-se ou aproximar-se do robô *Leader*.

O ângulo em que o robô *Helper* “vê” o robô *Leader*, isto é, a posição angular do robô *Leader* relativamente ao robô *Helper* é dado por:

$$\psi_{H,L} = atan2 \left( \frac{y_s + Bar \times \sin(\theta_e)}{x_s + Bar \times \cos(\theta_e)} \right) \quad (6.5)$$

onde  $(x_s, y_s)$  representam as coordenadas onde o suporte do objecto se encontra relativamente ao centro do robô *Helper*, *Bar* representa o comprimento do objecto e  $\theta_e$  representa o ângulo medido na junta rotacional (ver figura 6.1).

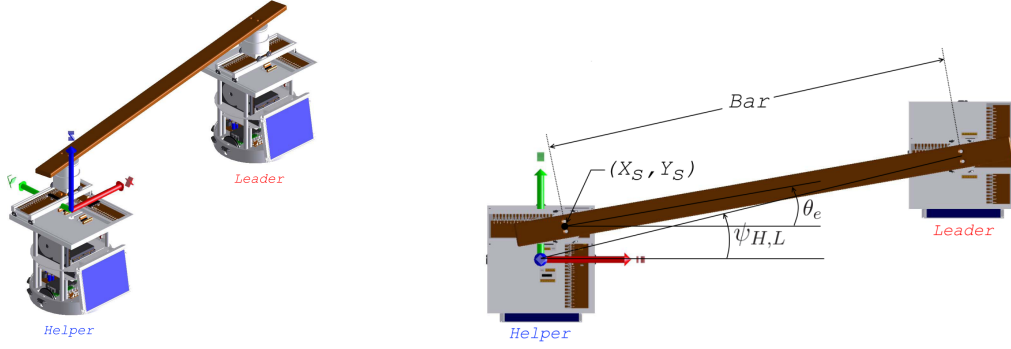


Figura 6.1:  $\theta_e$  representa o ângulo da junta rotacional, sendo nulo quando o robô *Helper* se encontra lado a lado, do lado esquerdo, do robô *Leader*.  $(x_s, y_s)$  representam as coordenadas do suporte do robô *Helper*, sendo este valor obtido pela leitura das juntas prismáticas do suporte do próprio robô.  $\psi_{H,L}$  é obtido através do ângulo  $\theta_e$  e das coordenadas  $(x_s, y_s)$ .

$$\Delta_{H_{any}} = \frac{2 \times \arctan_2(\alpha_{H_{any}} \times \Delta d_{H_{any}})}{\pi} \quad (6.6)$$

onde  $\Delta d_{H_{any}}$  representa a norma do deslocamento do objecto em relação ao centro do robô *Helper* em causa e  $\alpha_{H_{any}}$  é o factor de ganho da distância para o mesmo robô.

$$\Delta d_{H_{Any}} = \sqrt{x_{s,H_{any}}^2 + y_{s,H_{any}}^2} \quad (6.7)$$

em que  $(x_{s,H_{any}}, y_{s,H_{any}})$  representam as coordenadas onde o suporte do objecto se encontra no robô, em relação ao centro do próprio robô *Helper*.

Quanto maior for o valor do parâmetro  $\alpha_{H_{any}}$  mais rapidamente a função da equação 6.6 tende para a unidade. O valor de  $\alpha_{H_{any}}$  no qual se obteve melhores resultados no comportamento do robô *Helper* foi a unidade. Valores demasiado elevados fazem com que o robô *Helper* tenha um comportamento oscilatório, e valores demasiado pequenos fazem com que o robô *Helper* não consiga concluir a tarefa, isto é, não compense suficientemente o deslocamento do objecto no suporte, o que faz com que o robô *Helper* deixe cair o objecto.



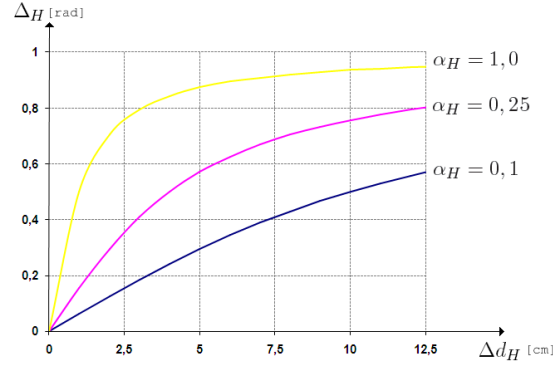


Figura 6.2: Representação da conversão da distância medida no suporte do objecto num ângulo entre 0 e 1 *rad*.

#### 6.1.1.1 Comportamento *Forward/Backward*

Este comportamento será o comportamento por defeito, isto é, quando o robô *Helper* estiver numa zona em que não detecte obstáculos deve ser este o comportamento activo.

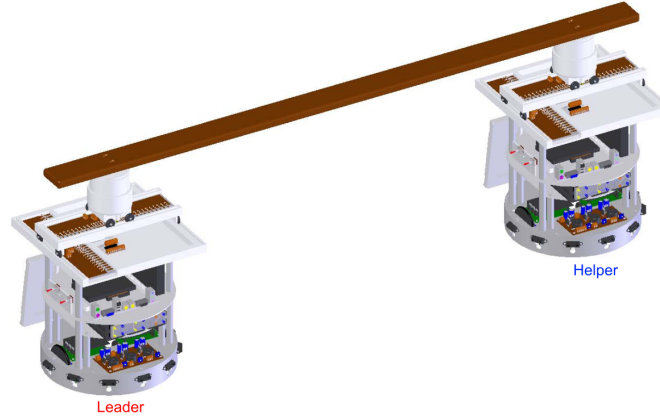


Figura 6.3: O robô *Helper* está no comportamento *Forward/Backward*. Este deve ser o comportamento activo quando este robô não detecte qualquer obstáculo.

Estando o robô *Helper* no comportamento *Forward/Backward*, este terá de manter-se lado a lado com o robô *Leader*.

Neste comportamento, *behaviour* (ver equação 6.4) corresponde ao comportamento *Forward/Backward*, ou seja, *behaviour* designa-se por *f/b*, assim tem-se que:

$$\psi_{desired,H,f/b} = \psi_{H,L} + \Delta\psi_{H,L,f/b} + R_{H,L,f/b} \times \Delta_H \quad (6.8)$$

em que  $\Delta\psi_{H,L,f/b}$  neste comportamento toma o valor de  $\frac{\pi}{2} \text{ rad}$  e os possíveis valores de  $R_{H,L,f/b}$  para o comportamento *Forward/Backward* são dados pela equação:

$$R_{H,L,f/b} = \begin{cases} 1 & , \text{ se } \alpha_{H,f} = 1 \\ -1 & , \text{ se } \alpha_{H,b} = 1 \end{cases} \quad (6.9)$$

onde  $\alpha_{H,b}$  e  $\alpha_{H,f}$  são variáveis de sinalização internas a este comportamento.

$$\alpha_{f,H} = \begin{cases} 1 & , \text{ se } \begin{aligned} & U_{H,obs}(\phi_H) \leq 0 \quad \wedge \\ & \overline{\theta_1 < \psi_{L,H} < \theta_2} \quad , \\ & \theta_1 < \theta_2 \quad , \quad \forall \{\theta_1, \theta_2\} \in [0, 2\pi[ \end{aligned} \\ 0 & , \text{ caso contrário} \end{cases} \quad (6.10)$$

em que  $U_{H,obs}(\phi_H)$  é a função potencial da dinâmica referente ao comportamento *evitar colisões com obstáculos*, dada de igual forma pela equação 5.19. Valores positivos indicam que a direcção de navegação do robô está dentro de uma região suficientemente forte, criada pela presença de obstáculos. Contrariamente, valores não positivos indicam que a direcção de navegação está numa zona de repulsão considerada fraca ou mesmo nula.

Interpretando a equação 6.10, tem-se que o robô *Helper* está em *Forward* se as obstruções detectadas pelo próprio robô são fracas/inexistentes e se o robô *Leader* não navegar numa direcção que vá de encontro ao robô *Helper*.

$$\alpha_{b,H} = \begin{cases} 1 & , \text{ se } \begin{aligned} & (\theta_1 < \psi_{L,H} < \theta_2) , \\ & \theta_1 < \theta_2 \quad , \quad \forall \{\theta_1, \theta_2\} \in [0, 2\pi[ \end{aligned} \\ 0 & , \text{ caso contrário} \end{cases} \quad (6.11)$$

Os valores de  $\theta_1$  e de  $\theta_2$  especificam o intervalo angular para o qual o robô *Helper* está

ou não com a componente *Backward*, do comportamento *Forward/Backward*, activada, que nesta dissertação tomam o valor de  $\frac{3\pi}{4} \text{ rad}$  e de  $\frac{5\pi}{4} \text{ rad}$ , respectivamente.

Interpretando a equação 6.11, tem-se que o robô *Helper* está em *Backward* sempre que o robô *Leader* navegue numa direcção que vá de encontro ao robô *Helper*.

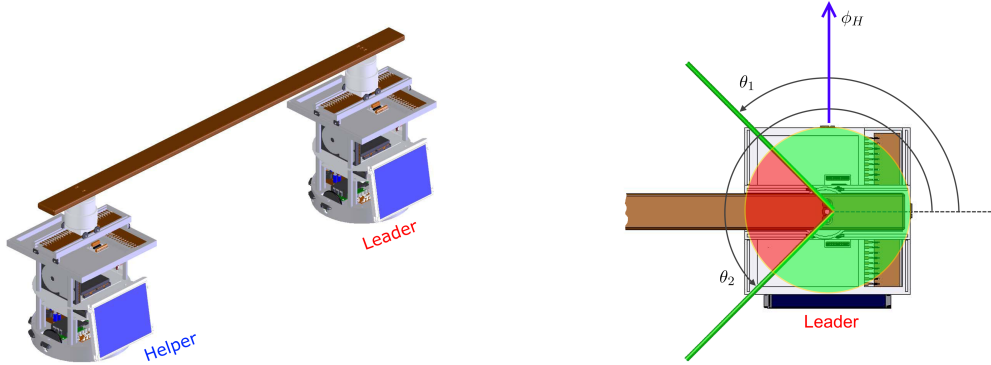


Figura 6.4: Regiões em que a direcção de navegação do robô *Leader* faz com que o robô *Helper* esteja em *Forward*, região a verde, ou em *Backward*, região a vermelho, sendo o comportamento *Forward/Backward* o activo.

Caso se verifique um valor positivo das equações 6.10 ou 6.11 então o comportamento *Forward/Backward* é activado:

$$\gamma_{H,f/b} = \begin{cases} 1 & , \text{ se } \alpha_{H,f} = 1 \vee \alpha_{H,b} = 1 \\ 0 & , \text{ caso contrário} \end{cases} \quad (6.12)$$

Por construção do suporte e *setup* da equipa, sabe-se que o robô *Leader* quando está lado a lado com o robô *Helper* (do lado esquerdo), no comportamento *Forward/Backward*, isto é, como na condição da figura 6.3, a junta rotacional do robô *Leader* está a medir o valor  $\pi \text{ rad}$  (isto porque se considera que o robô *Helper* fica do lado esquerdo). Assim sendo, sabe-se que, se a junta rotacional indicar valores entre  $(\frac{3\pi}{2} - \theta_1) \text{ rad}$  e  $(\frac{3\pi}{2} - \theta_2) \text{ rad}$ , então significa que a direcção de navegação do robô *Leader* aponta para a região a vermelho da figura 6.4(b), o que fará com que o robô *Helper* caso esteja no comportamento *Forward/Backward* esteja em *Backward*, caso contrário, fará com que esteja em *Forward*.

### 6.1.1.2 Comportamento *Turn*

Este comportamento é activado quando o robô *Helper* apenas detecta obstáculos de um dos lados.

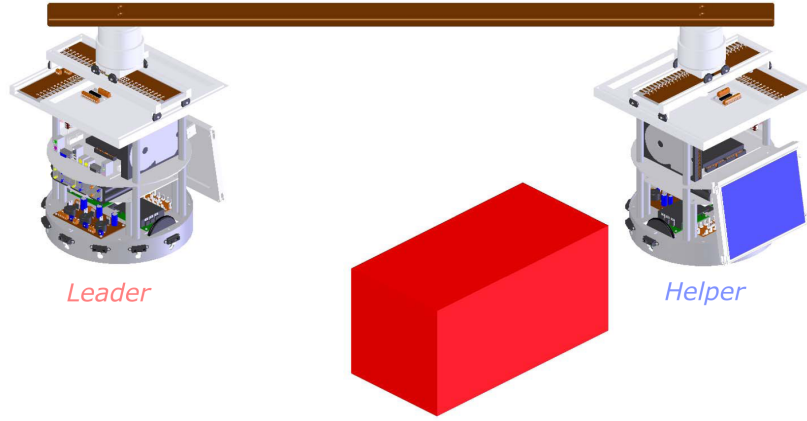


Figura 6.5: O robô *Helper* no comportamento *Turn*. Este comportamento deve ser activado quando este detecta obstáculos apenas de um dos lados.

Estando o robô *Helper* no comportamento *Turn*, este terá de rodar para a esquerda ou para a direita por forma a evitar colidir com o(s) obstáculo(s).

Neste comportamento, *behaviour* (ver equação 6.4) corresponde ao comportamento *Turn*, ou seja, *behaviour* designa-se por  $t$ , assim tem-se que:

$$\psi_{desired,H,t} = \psi_{H,L} + \Delta\psi_{H,L,t} + R_{H,L,t} \times \Delta_H \quad (6.13)$$

em que  $\Delta\psi_{H,L,t}$  neste comportamento toma o valor de  $\pm\frac{\pi}{4} \text{ rad}$  e  $R_{H,L,t}$  toma o valor de  $\pm 1$ , dependendo do lado em que são sentidos as obstruções pelo robô, como se pode verificar nas duas equações que se seguem.

$$\Delta\psi_{H,L,t} = \begin{cases} +\frac{\pi}{4} \text{ rad} & , \text{ se } \alpha_{H,tl} = 1 \\ -\frac{\pi}{4} \text{ rad} & , \text{ se } \alpha_{H,tr} = 1 \end{cases} \quad (6.14)$$

$$R_{H,L,t} = \begin{cases} +1 & , \text{ se } \alpha_{H,tl} = 1 \\ -1 & , \text{ se } \alpha_{H,tr} = 1 \end{cases} \quad (6.15)$$

As variáveis  $\alpha_{H,tl}$  e  $\alpha_{H,tr}$  são variáveis internas ao comportamento, que dizem respeito à viragem à esquerda e à direita, respectivamente.

$$\alpha_{tl} = \begin{cases} 1 & , \text{ se } \begin{aligned} & U_{H,obs}(\phi_H) > 0 \wedge f_{H,obs}(\phi_H) > 0 \quad \wedge \\ & |\psi_{H,L} - \frac{\pi}{2}| > \theta_3 \end{aligned} \\ 0 & , \text{ caso contrário} \end{cases} \quad (6.16)$$

$$\alpha_{tr} = \begin{cases} 1 & , \text{ se } \begin{aligned} & U_{H,obs}(\phi_H) > 0 \wedge f_{H,obs}(\phi_H) \leq 0 \quad \wedge \\ & |\psi_{H,L} - \frac{\pi}{2}| > \theta_3 \end{aligned} \\ 0 & , \text{ caso contrário} \end{cases} \quad (6.17)$$

em que  $U_{H,obs}(\phi_H)$  é a função potencial da dinâmica referente ao campo vectorial da dinâmica *evitar colisões com obstáculos*, dada de igual forma pela equação 5.19.  $f_{H,obs}(\phi_H)$  é o campo vectorial resultante da dinâmica *evitar colisões com obstáculos*, dada pela equação 5.10. Valores positivos indicam que são sentidas obstruções do lado direito do robô, enquanto que valores negativos indicam que as obstruções são sentidas do lado esquerdo.  $\theta_3$  é um parâmetro que indica se o robô está ou não na gama angular do comportamento *Column*, tendo sido definido igual a  $\frac{\pi}{18} \text{ rad}$ .

Interpretando a equação 6.16, o robô *Helper vira à esquerda* quando este detecta obstruções, essas obstruções são sentidas apenas do lado direito do robô e a diferença da direcção de navegação com a orientação do objecto a transportar for maior que um determinado ângulo,  $\theta_3$ .

Interpretando agora a equação 6.17, o robô *Helper vira à direita* quando este detecta obstruções, essas obstruções são sentidas apenas do lado esquerdo do robô e a diferença

da direcção de navegação com a orientação do objecto a transportar for maior que um determinado ângulo,  $\theta_3$ .

Caso se verifique um valor positivo das equações 6.16 ou 6.17 então o comportamento *Turn* é activado, como se pode ver pela equação:

$$\gamma_t = \begin{cases} 1 & , \text{ se } \alpha_{H,tl} = 1 \vee \alpha_{H,tr} = 1 \\ 0 & , \text{ caso contrário} \end{cases} \quad (6.18)$$

### 6.1.1.3 Comportamento *Column*

Na presença de obstáculos longos ou em passagens estreitas, onde obstruções são detetadas de ambos os lados, figura 6.6, o robô *Helper* tem de navegar atrás do robô *Leader* (i.e., navegar mantendo o comportamento *Column*).

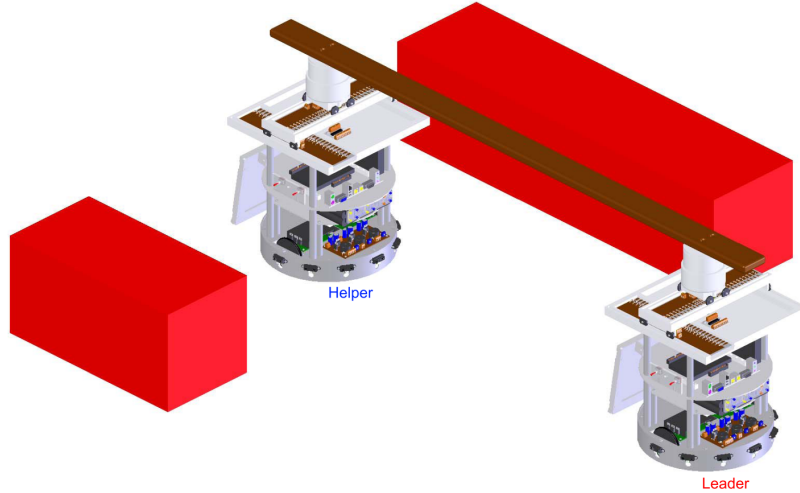


Figura 6.6: Direcção desejada para o robô *Helper* no comportamento *Column*.

Neste comportamento, *behaviour* (ver equação 6.4) corresponde ao comportamento *Column*, ou seja, *behaviour* designa-se por *c*, assim tem-se que:

$$\psi_{desired,H,c} = \psi_{H,L} + \Delta\psi_{H,L,c} + R_{H,L,c} \times \Delta_H \quad (6.19)$$

em que  $\Delta\psi_{H,L,c}$  neste comportamento toma o valor de  $0 \text{ rad}$  e  $R_{H,L,c}$  também toma o valor de 0, pois se o robô se encontra atrás do robô *Leader*, o suporte do objecto só sofrerá deslocamento segundo *Oy*, o que será compensado através da velocidade.

$$\gamma_c = \begin{cases} 1 & , \text{ se } U_{H,obs}(\phi_H) > 0 \wedge |\psi_{H,L} - \frac{\pi}{2}| \leq \theta_3 \\ 0 & , \text{ caso contrário} \end{cases} \quad (6.20)$$

em que  $U_{H,obs}(\phi_H)$  é a função potencial da dinâmica referente ao campo vectorial da dinâmica *evitar colisões com obstáculos*, dada de igual forma pela equação 5.19.  $\theta_3$  é um parâmetro, que foi obtido por simulação, e que toma o valor de  $\frac{\pi}{18} \text{ rad}$ . Este ângulo é uma janela para o comportamento *Column*, isto é, para que o comportamento *Column* não seja activado apenas quando o robô *Helper* se encontre mesmo a  $270^\circ$  relativamente ao robô *Leader*.

Interpretando a equação 6.20, o robô *Helper* fica com o comportamento *Column* activo quando o próprio detecta obstruções e a diferença da direcção de navegação com a orientação do objecto a transportar for menor ou igual a um determinado ângulo,  $\theta_3$ , seguindo assim atrás do robô *Leader*.

### 6.1.2 Para a equipa de três robôs

Para cada comportamento, o ângulo desejado segundo o qual o atractor da dinâmica, para cada robô *Helper*, deve ser erigido, é dado pela equação:

$$\psi_{desired,H_i,behaviour} = \frac{\psi_{desired,H_i,L,behaviour} + \psi_{desired,H_i,H_j,behaviour}}{2}, \quad (6.21)$$

$$i = \{1, 2\}, \quad j = \{1, 2\} \wedge j \neq i$$

em que

$$\begin{aligned} \psi_{desired,H_i,L,behaviour} &= \psi_{H_i,L} + \Delta\psi_{H_i,L,behaviour} + R_{H_i,L,behaviour} \times \Delta_{H_i} \quad , \\ i &= \{1, 2\} \end{aligned} \quad (6.22)$$

$$\begin{aligned} \psi_{desired,H_i,H_j,behaviour} &= \psi_{H_i,H_j} + \Delta\psi_{H_i,H_j,behaviour} + R_{H_i,H_j,behaviour} \times \Delta_{H_i} \quad , \\ i &= \{1, 2\}, j = \{1, 2\} \wedge j \neq i \end{aligned} \quad (6.23)$$

$\psi_{desired,H_i,L,behaviour}$  representa a direcção de navegação desejada para o robô  $H_i$  (*Helper 1* ou *Helper 2*) em relação ao robô *Leader* para o comportamento especificado por *behaviour*.  $\psi_{desired,H_i,H_j,behaviour}$  representa a direcção de navegação desejada para o robô  $H_i$  (*Helper 1* ou *Helper 2*) em relação ao robô  $H_j$  (*Helper 2* ou *Helper 1*) com  $j \neq i$ , para o comportamento especificado por *behaviour*.

$\psi_{H_i,L}$  representa o ângulo em que o robô  $H_i$  (*Helper 1* ou *Helper 2*) “vê” o robô *Leader*.  $\psi_{H_i,H_j}$  representa o ângulo em que o robô  $H_i$  (*Helper 1* ou *Helper 2*) “vê” o robô  $H_j$  (*Helper 2* ou *Helper 1*) com  $j \neq i$ .

$\Delta\psi_{H_i,L,behaviour}$  especifica o ângulo característico do robô  $H_i$  (*Helper 1* ou *Helper 2*) em relação ao robô *Leader*, para o comportamento especificado por *behaviour*. O ângulo característico do robô  $H_i$  (*Helper 1* ou *Helper 2*) em relação ao robô  $H_j$  (*Helper 2* ou *Helper 1*) com  $j \neq i$ , para o comportamento especificado por *behaviour*, é especificado por  $\Delta\psi_{H_i,H_j,behaviour}$ .

$\Delta_{H_i}$  representa um ângulo que é proporcional ao deslocamento que o robô  $H_i$  (*Helper 1* ou *Helper 2*) “sente” no seu suporte do objecto (ver equação 6.6). Também para esta equipa, o valor de  $\alpha_{H_{any}}$  que apresentou melhor comportamento foi o mesmo que o utilizado no *Helper* da equipa de dois robôs, isto é, a unidade ( $\alpha_{H_{any}} = 1$ ).

$R_{H_i,L,behaviour}$  é um parâmetro que indica ao sistema de controlo se o robô  $H_i$  (*Helper 1* ou *Helper 2*) está a uma distância maior ou menor que a distância desejada ao robô



*Leader* fazendo com que  $\Delta_{H_i}$  seja somado ou subtraído, para que o robô  $H_i$  (*Helper 1* ou *Helper 2*) se afaste ou se aproxime do robô *Leader*, respectivamente.  $R_{H_i,H_j,behaviour}$  é um parâmetro que indica ao sistema de controlo se o robô  $H_i$  (*Helper 1* ou *Helper 2*) está a uma distância maior ou menor que a distância desejada ao robô  $H_j$  (*Helper 2* ou *Helper 1*) com  $j \neq i$ , fazendo com que  $\Delta_{H_i}$  seja somado ou subtraído, para que o robô  $H_i$  (*Helper 1* ou *Helper 2*) se afaste ou se aproxime do robô  $H_j$  (*Helper 2* ou *Helper 1*) com  $j \neq i$ , respectivamente.

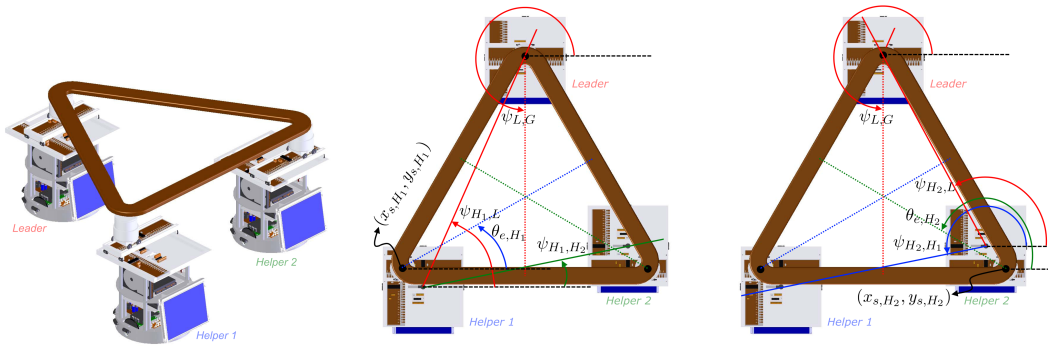


Figura 6.7:  $\theta_{e,H_i}$  representa o ângulo da junta rotacional.  $(x_{s,H_i}; y_{s,H_i})$  representam as coordenadas do suporte em que o objecto a transportar se encontra deslocado do centro do robô  $H_i$ .  $\psi_{H_i,L}$  é obtido através do ângulo  $\theta_{e,H_i}$  e das coordenadas  $(x_{s,H_i}; y_{s,H_i})$ . Para cada variável,  $i = \{1, 2\}$ .

### 6.1.2.1 Comportamento *Forward/Backward*

Neste comportamento, *behaviour* (ver equações 6.21, 6.22 e 6.23) corresponde ao comportamento *Forward/Backward*, ou seja, *behaviour* designa-se por *f/b*, assim tem-se que:

$$\psi_{desired,H_i,L,f/b} = \psi_{H_i,L} + \Delta\psi_{H_i,L,f/b} + R_{H_i,L,f/b} \times \Delta_{H_i} \quad (6.24)$$

$$\psi_{desired,H_i,H_j,f/b} = \psi_{H_i,H_j} + \Delta\psi_{H_i,H_j,f/b} + R_{H_i,H_j,f/b} \times \Delta_{H_i} \quad (6.25)$$

No comportamento *Forward/Backward* os valores dos parâmetros  $\Delta\psi_{H_i,L,behaviour}$  e

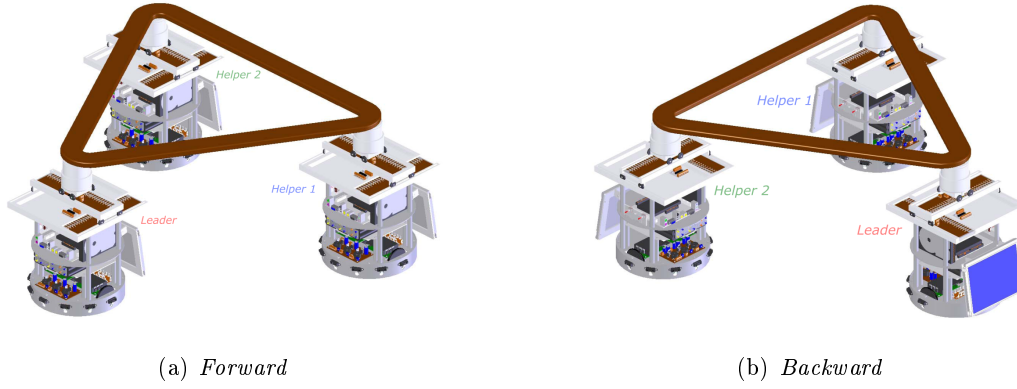


Figura 6.8: Equipa de 3 robôs no comportamento *Forward/Backward*. Na imagem (a) está representada a acção *Forward*, e na imagem (b) a acção *Backward*.

$\Delta\psi_{H_i,H_j,behaviour}$ , isto é,  $\Delta\psi_{H_i,L,f/b}$  e  $\Delta\psi_{H_i,H_j,f/b}$ , podem ser visualizados na figura 6.9, e são dados por:

$$\Delta\psi_{H_i,L,f/b} = \begin{cases} +\frac{\pi}{6} \text{ rad} & , \text{ se } i = 1 \\ -\frac{\pi}{6} \text{ rad} & , \text{ se } i = 2 \end{cases} \quad (6.26)$$

$$\Delta\psi_{H_i,H_j,f/b} = \begin{cases} +\frac{\pi}{2} \text{ rad} & , \text{ se } i = 1 \wedge j = 2 \\ -\frac{\pi}{2} \text{ rad} & , \text{ se } i = 2 \wedge j = 1 \end{cases} \quad (6.27)$$

Os valores dos ângulos referidos fazem com que nesta formação os robôs mantenham uma configuração tipo triângulo equilátero, para qualquer outro tipo de configuração geométrico ter-se-ia que alterar estes valores.

Os valores de  $R_{H_i,L,f/b}$  e  $R_{H_i,H_j,f/b}$  para o comportamento *Forward/Backward* são dados por:

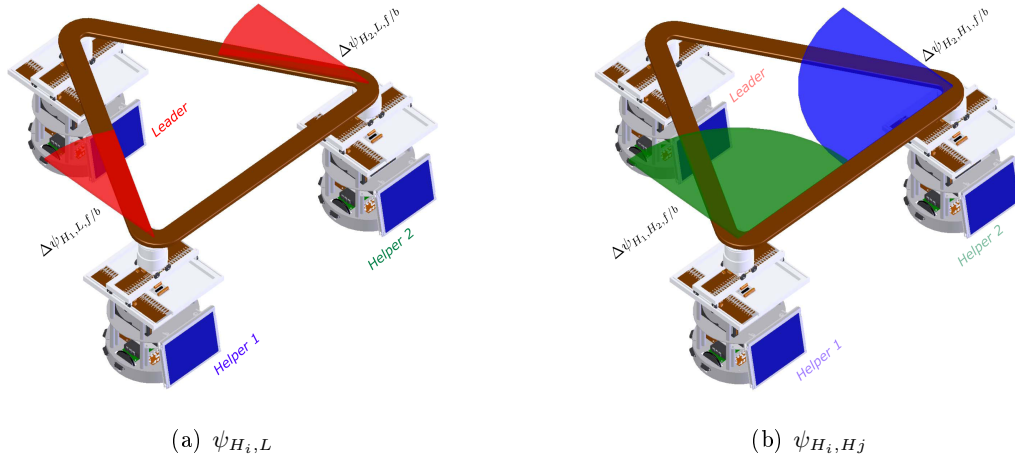


Figura 6.9: Representação dos ângulos característicos para o comportamento *Forward/Backward*, isto é, de  $\Delta\psi_{H_i,L,f/b}$  e de  $\Delta\psi_{H_i,H_j,f/b}$ , com  $i = \{1, 2\}$  e  $j = \{1, 2\} \wedge j \neq i$  para o comportamento *Forward/Backward*.

$$R_{H_i,L,f/b} = \begin{cases} +1 & , \text{ se } i = 1 \wedge \alpha_{H_i,f} = 1 \\ -1 & , \text{ se } i = 2 \wedge \alpha_{H_i,f} = 1 \\ -1 & , \text{ se } i = 1 \wedge \alpha_{H_i,b} = 1 \\ +1 & , \text{ se } i = 2 \wedge \alpha_{H_i,b} = 1 \end{cases} \quad (6.28)$$

$$R_{H_i,H_j,f/b} = \begin{cases} +1 & , \text{ se } i = 1 \wedge j = 2 \wedge \alpha_{H_i,f} = 1 \\ -1 & , \text{ se } i = 2 \wedge j = 1 \wedge \alpha_{H_i,f} = 1 \\ -1 & , \text{ se } i = 1 \wedge j = 2 \wedge \alpha_{H_i,b} = 1 \\ +1 & , \text{ se } i = 2 \wedge j = 1 \wedge \alpha_{H_i,b} = 1 \end{cases} \quad (6.29)$$

onde  $\alpha_{H_i,f}$  e  $\alpha_{H_i,b}$  são variáveis de sinalização internas referentes ao comportamento *Forward/Backward* do robô  $H_i$  (*Helper 1* ou *Helper 2*).

As variáveis de sinalização internas referentes ao comportamento aqui descrito,  $\alpha_{H_i,f}$  e  $\alpha_{H_i,b}$ , são dadas por:

$$\alpha_{H_i,f} = \begin{cases} 1 & , \text{ se } \left[ \begin{aligned} & (U_{H_i,obs}(\phi_{H_i}) \leq 0 \wedge U_{H_j,obs}(\phi_{H_j}) \leq 0) \vee \\ & (U_{H_i,obs}(\phi_{H_i}) > 0 \wedge U_{H_j,obs}(\phi_{H_j}) > 0) \end{aligned} \right] \wedge \\ & \frac{\psi_{L,G} > \frac{5\pi}{4} \wedge \psi_{L,G} < \frac{7\pi}{4}}{i = \{1, 2\}, \quad j = \{1, 2\} \wedge j \neq i} \quad , \quad (6.30) \\ 0 & , \text{ caso contrário} \end{cases}$$

$$\alpha_{H_i,b} = \begin{cases} 1 & , \text{ se } \psi_{L,G} > \frac{5\pi}{4} \wedge \psi_{L,G} < \frac{7\pi}{4} \quad , \\ 0 & , \text{ caso contrário} \end{cases} \quad (6.31)$$

em que  $U_{H_i,obs}$  é a função potencial da dinâmica referente ao comportamento *evitar colisões com obstáculos*, dada de igual forma pela equação 5.19.  $\psi_{L,G}$  especifica a posição angular em que se encontra o restante grupo em relação ao robô *Leader*, este valor, por *setup* da equipa, quando esta se encontra como na figura 6.8(a) corresponde a  $\frac{3\pi}{2}$  *rad*. Os valores de  $\theta_1$  e de  $\theta_2$  (ver figura 6.10(b)) especificam o intervalo angular para o qual os robôs *Helpers* estão ou não com a componente *Backward*, do comportamento *Forward/Backward*, activada, que nesta dissertação tomam o valor de  $\frac{\pi}{4}$  *rad* e de  $\frac{3\pi}{4}$  *rad*, respectivamente. Estes valores foram determinados por simulação. Assim sendo, sabe-se que, se a junta rotacional indicar valores entre  $(\pi + \theta_1)$  *rad* e  $(\pi + \theta_2)$  *rad*, então significa que a direcção de navegação do robô *Leader* aponta para a região a verde da figura 6.10(b), o que fará com que o robô *Helper* caso esteja no comportamento *Forward/Backward* esteja em *Forward*, caso contrário, fará com que esteja em *Backward*.

Interpretando a equação 6.30, tem-se que o robô  $H_i$  (*Helper 1* ou *Helper 2*) está em *Forward* se as obstruções detectadas pelos próprios robôs  $H_i$  são fracas/inexistentes e se o robô *Leader* não navegar numa direcção que vá de encontro a um dos robôs *Helpers*.

Interpretando a equação 6.31, tem-se que os robôs *Helpers* estão em *Backward* sempre que o robô *Leader* navegue numa direcção que vá de encontro a um dos robôs *Helpers*.

Caso se verifique um valor positivo das equações 6.30 ou 6.31 então o comportamento *Forward/Backward* é activado:

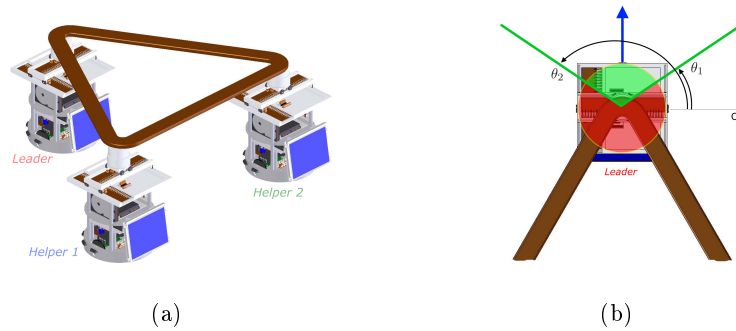


Figura 6.10: Regiões em que a direcção de navegação do robô *Leader* faz com que o robô *Helper* esteja em *Forward*, região a verde, ou em *Backward*, região a vermelho, sendo o comportamento *Forward/Backward* o activo.

$$\gamma_{H,f/b} = \begin{cases} 1 & , \text{ se } \alpha_{H_i,f} = 1 \vee \alpha_{H_i,b} = 1, \quad i = \{1,2\} \\ 0 & , \text{ caso contrário} \end{cases} \quad (6.32)$$

Por construção do suporte e *setup* da equipa, sabe-se que os robôs  $H_i$  (*Helper 1* e *Helper 2*) estão lado a lado no comportamento *Forward/Backward*, isto é, como na condição da figura 6.8. A junta rotacional do robô *Leader*, na figura 6.8(a), indica o valor  $\frac{3\pi}{2} \text{ rad}$ , e na figura 6.8(b), indica o valor  $\frac{\pi}{2} \text{ rad}$ . O valor desta junta é comunicado aos robôs *Helpers*, ficando estes assim a saber qual a posição angular da equipa em relação ao robô *Leader*. A junta rotacional do robô *Helper 1* indica o valor  $\frac{\pi}{6} \text{ rad}$ , este valor indica a posição angular da equipa em relação ao suporte do robô *Helper 1*. Por último, a junta rotacional do robô *Helper 2* indica o valor  $\frac{5\pi}{6} \text{ rad}$ , este valor indica a posição angular da equipa em relação ao suporte do robô *Helper 2*.

#### 6.1.2.2 Comportamento *Turn*

Este comportamento é activado apenas quando um dos robôs  $H_i$  (*Helper 1* ou *Helper 2*) detecta obstáculos, motivo pelo qual é necessário a comunicação bidirecional entre os dois robôs *Helpers*. A informação comunicada entre os robôs *Helpers* são o valor função da dinâmica *evitar colisões com obstáculos* e a respectiva função potencial. É de salientar que para os robôs *Helpers* a função da dinâmica *evitar colisões com obstáculos*

apenas tem significado para a activação de comportamentos, ou seja, o que faz com que os robôs *Helpers* evitem colidir com os obstáculos é a alteração de comportamento, pois a função da dinâmica *evitar colisões com obstáculos* não influencia directamente a direcção de navegação desejada.

Estando os robôs *Helpers* com o comportamento *Turn* activo, estes terão de rodar para a esquerda ou para a direita, por forma a evitar que colidam com obstáculos.

Neste comportamento, *behaviour* (ver equações 6.21, 6.22 e 6.23) corresponde ao comportamento *Turn*, ou seja, *behaviour* designa-se por  $t$ , assim tem-se que:

$$\psi_{desired,H_i,L,t} = \psi_{H_i,L} + \Delta\psi_{H_i,L,t} + R_{H_i,L,t} \times \Delta_{H_i} \quad (6.33)$$

$$\psi_{desired,H_i,H_j,t} = \psi_{H_i,H_j} + \Delta\psi_{H_i,H_j,t} + R_{H_i,H_j,t} \times \Delta_{H_i} \quad (6.34)$$

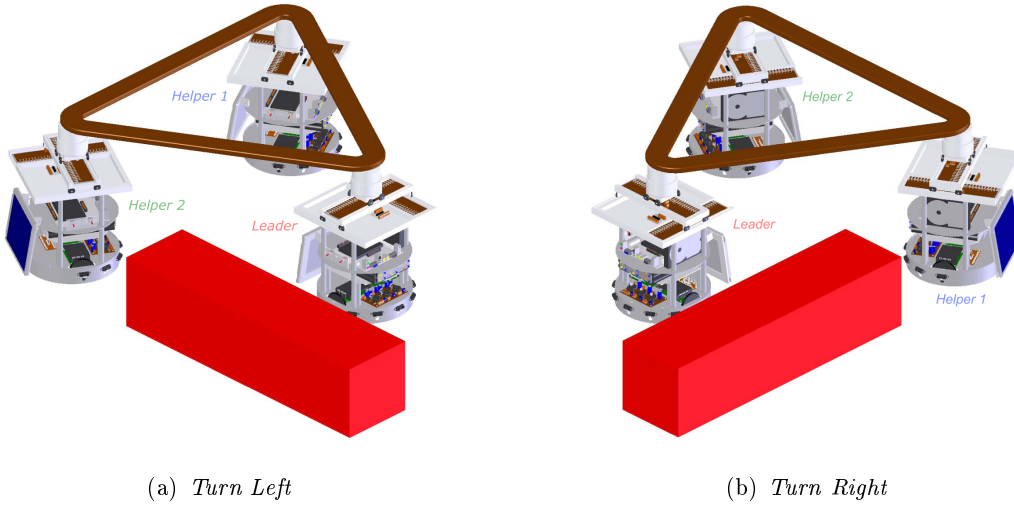


Figura 6.11: Equipa de 3 robôs no comportamento *Turn*. Na imagem (a) está representada a acção *Left*, e na imagem (b) a acção *Right*.

No comportamento *Turn*,  $\Delta\psi_{H_i,L}$  e  $\Delta\psi_{H_i,H_j}$ , isto é,  $\Delta\psi_{H_i,L,t}$  e  $\Delta\psi_{H_i,H_j,t}$  são dados pelas equações 6.35 e 6.36, e representados nas figuras 6.12 e 6.13, respectivamente.

$$\Delta\psi_{H_i,L,t} = \begin{cases} +\frac{\pi}{3} \text{ rad} & , \text{ se } i = 1 \wedge \alpha_{H_i,tl} = 1 \\ +\frac{\pi}{6} \text{ rad} & , \text{ se } i = 2 \wedge \alpha_{H_i,tl} = 1 \\ -\frac{\pi}{6} \text{ rad} & , \text{ se } i = 1 \wedge \alpha_{H_i,tr} = 1 \\ -\frac{\pi}{3} \text{ rad} & , \text{ se } i = 2 \wedge \alpha_{H_i,tr} = 1 \end{cases} \quad (6.35)$$

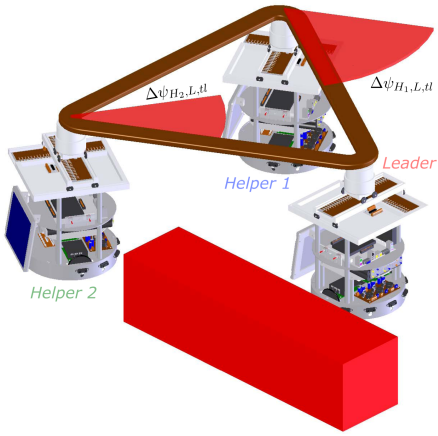
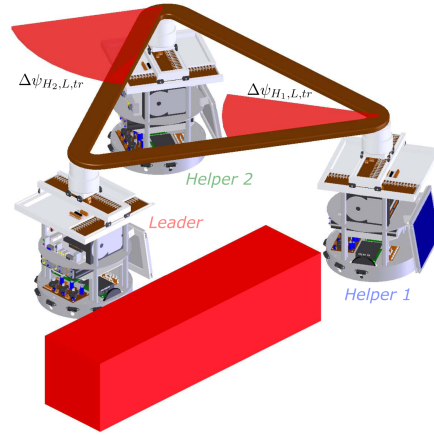
(a) Representação de  $\Delta\psi_{H_i,L,tl}$ (b) Representação de  $\Delta\psi_{H_i,L,tr}$ 

Figura 6.12: Representação dos ângulos característicos dos robôs *Helpers* relativamente ao robô *Leader* para o comportamento *Turn*, isto é, de  $\Delta\psi_{H_i,L,t}$ , com  $i = \{1, 2\}$ .

$$\Delta\psi_{H_i,H_j,t} = \begin{cases} +\frac{2\pi}{3} \text{ rad} & , \text{ se } i = 1 \wedge j = 2 \wedge \alpha_{H_i,tl} = 1 \\ -\frac{\pi}{6} \text{ rad} & , \text{ se } i = 2 \wedge j = 1 \wedge \alpha_{H_i,tl} = 1 \\ +\frac{\pi}{6} \text{ rad} & , \text{ se } i = 1 \wedge j = 2 \wedge \alpha_{H_i,tr} = 1 \\ -\frac{2\pi}{3} \text{ rad} & , \text{ se } i = 2 \wedge j = 1 \wedge \alpha_{H_i,tr} = 1 \end{cases} \quad (6.36)$$

Os valores de  $R_{H_i,L}$  e  $R_{H_i,H_j}$  para o comportamento *Turn*, isto é,  $R_{H_i,L,t}$  e  $R_{H_i,H_j,t}$  são dados por:

$$R_{H_i,L,t} = \begin{cases} +1 & , \text{ se } i = 1 \\ -1 & , \text{ se } i = 2 \end{cases} \quad (6.37)$$

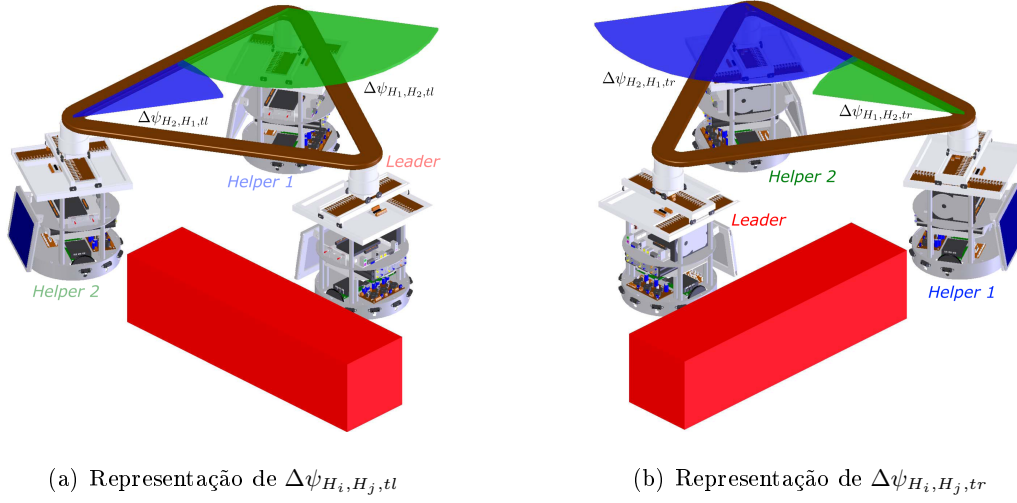


Figura 6.13: Representação dos ângulos característicos para o robô  $H_i$  relativamente ao robô  $H_j$  para o comportamento *Turn*, isto é, de  $\Delta\psi_{H_i, H_j, t}$ , com  $i = \{1, 2\}$  e  $j = \{1, 2\} \wedge j \neq i$ .

$$R_{H_i, H_j, t} = \begin{cases} +1 & , \text{ se } i = 1 \wedge j = 2 \\ -1 & , \text{ se } i = 2 \wedge j = 1 \end{cases} \quad (6.38)$$

As variáveis  $\alpha_{H_i, tl}$  e  $\alpha_{H_i, tr}$  são de sinalização ao comportamento *Turn* e correspondem à viragem à esquerda e à direita, respectivamente, por forma a que os robôs *Helpers* evitem o obstáculo.

$$\alpha_{H_i, tl} = \begin{cases} 1 & , \text{ se } \left[ (U_{H_1, obs}(\phi_{H_1}) > 0 \wedge f_{H_1, obs}(\phi_{H_1}) > 0) \vee \right. \\ & (U_{H_2, obs}(\phi_{H_2}) > 0 \wedge f_{H_2, obs}(\phi_{H_2}) > 0) \bigg] \wedge \\ & \left[ \overline{(U_{H_1, obs}(\phi_{H_1}) > 0 \wedge f_{H_1, obs}(\phi_{H_1}) > 0)} \wedge \right. \\ & \left. \overline{(U_{H_2, obs}(\phi_{H_2}) > 0 \wedge f_{H_2, obs}(\phi_{H_2}) > 0)} \right] \wedge & , \quad i = \{1, 2\} \\ & |\psi_{H_i, L} - \frac{\pi}{2}| > \theta_3 \\ 0 & , \text{ se não} \end{cases} \quad (6.39)$$



$$\alpha_{H_i,tr} = \begin{cases} 1 & , \text{ se } \left[ \overline{(U_{H_1,obs}(\phi_{H_1}) > 0 \wedge f_{H_1,obs}(\phi_{H_1}) > 0)} \wedge \right. \\ & \left. \overline{(U_{H_2,obs}(\phi_{H_2}) > 0 \wedge f_{H_2,obs}(\phi_{H_2}) > 0)} \right] \wedge \\ & [(U_{H_1,obs}(\phi_{H_1}) > 0 \wedge f_{H_1,obs}(\phi_{H_1}) > 0) \vee \\ & (U_{H_2,obs}(\phi_{H_2}) > 0 \wedge f_{H_2,obs}(\phi_{H_2}) > 0)] \wedge \\ & |\psi_{H_i,L} - \frac{\pi}{2}| > \theta_3 \\ 0 & , \text{ se não} \end{cases} \quad , \quad i = \{1, 2\} \quad (6.40)$$

em que  $U_{H_i,obs}$  é a função potencial da dinâmica referente ao campo vectorial da dinâmica *evitar colisões com obstáculos*, dada de igual forma pela equação 5.19.  $f_{H_i,obs}$  é o campo vectorial resultante da dinâmica *evitar colisões com obstáculos*, dada pela equação 5.10.  $\theta_3$  é um parâmetro, o qual foi definido como sendo  $\frac{\pi}{18} \text{ rad}$ . Este valor foi determinado por simulação.  $\psi_{H_i,L}$  representa o ângulo segundo o qual o robô  $H_i$  (*Helper 1* ou *Helper 2*) “vê” o robô *Leader*.

Caso se verifique um valor positivo das equações 6.39 ou 6.40 então o comportamento *Turn* é activado:

$$\gamma_{H_i,t} = \begin{cases} 1 & , \text{ se } \alpha_{H_i,tl} = 1 \vee \alpha_{H_i,tr} = 1, \quad i = \{1, 2\} \\ 0 & , \text{ caso contrário} \end{cases} \quad (6.41)$$

### 6.1.2.3 Comportamento *Column*

Estando os robôs *Helpers* com o comportamento *Column* activo, estes terão que navegar paralelamente ao obstáculo(s).

Neste comportamento, *behaviour* (ver equações 6.21, 6.22 e 6.23) corresponde ao comportamento *Column*, ou seja, *behaviour* designa-se por *c*, assim tem-se que:

$$\psi_{desired,H_i,L,c} = \psi_{H_i,L} + \Delta\psi_{H_i,L,c} + R_{H_i,L,c} \times \Delta_{H_i} \quad (6.42)$$

$$\psi_{desired,H_i,H_j,c} = \psi_{H_i,H_j} + \Delta\psi_{H_i,H_j,c} + R_{H_i,H_j,c} \times \Delta_{H_i} \quad (6.43)$$

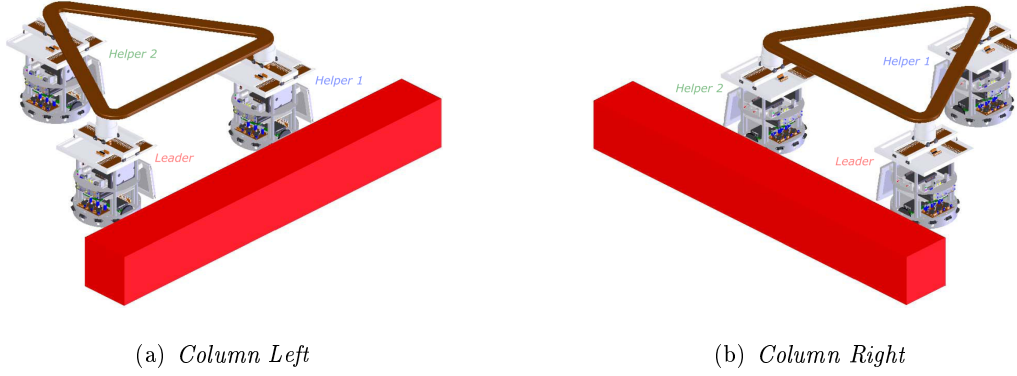


Figura 6.14: Equipa de 3 robôs no comportamento *Column*. Na imagem (a) está representada a activação da sinalização *Left*, e na imagem (b) a sinalização *Right*.

No comportamento *Column*  $\Delta\psi_{H_i,L}$  e  $\Delta\psi_{H_i,H_j}$ , isto é,  $\Delta\psi_{H_i,L,c}$  e  $\Delta\psi_{H_i,H_j,c}$  são dados pelas equações 6.44 e 6.45, e representados nas figuras 6.15 e 6.16, respectivamente.

$$\Delta\psi_{H_i,L,c} = \begin{cases} 0 \text{ rad} & , \text{ se } i = 1 \wedge \alpha_{H_i,cl} = 1 \\ -\frac{\pi}{3} \text{ rad} & , \text{ se } i = 2 \wedge \alpha_{H_i,cl} = 1 \\ +\frac{\pi}{3} \text{ rad} & , \text{ se } i = 1 \wedge \alpha_{H_i,cr} = 1 \\ 0 \text{ rad} & , \text{ se } i = 2 \wedge \alpha_{H_i,cr} = 1 \end{cases} \quad (6.44)$$

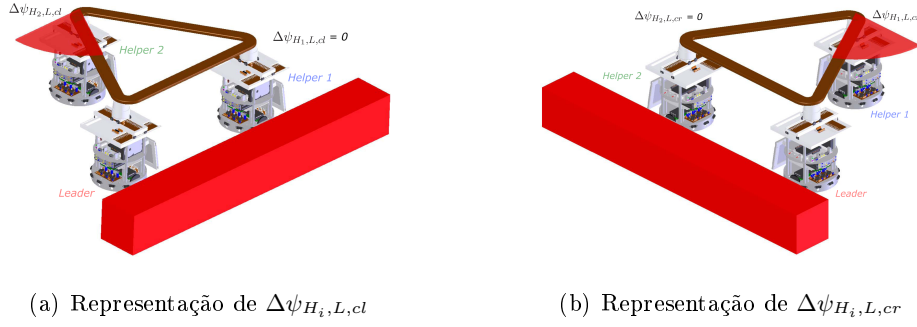


Figura 6.15: Representação dos ângulos característicos do robôs *Helpers* relativamente ao robô *Leader* para o comportamento *Column*, isto é, de  $\Delta\psi_{H_i,L,c}$ , com  $i = \{1, 2\}$ .

$$\Delta\psi_{H_i,H_j,c} = \begin{cases} +\frac{\pi}{3} \text{ rad} & , \text{ se } i = 1 \wedge j = 2 \wedge \alpha_{H_i,cl} = 1 \\ -\frac{2\pi}{3} \text{ rad} & , \text{ se } i = 2 \wedge j = 1 \wedge \alpha_{H_i,cl} = 1 \\ +\frac{2\pi}{3} \text{ rad} & , \text{ se } i = 1 \wedge j = 2 \wedge \alpha_{H_i,cr} = 1 \\ -\frac{\pi}{3} \text{ rad} & , \text{ se } i = 2 \wedge j = 1 \wedge \alpha_{H_i,cr} = 1 \end{cases} \quad (6.45)$$

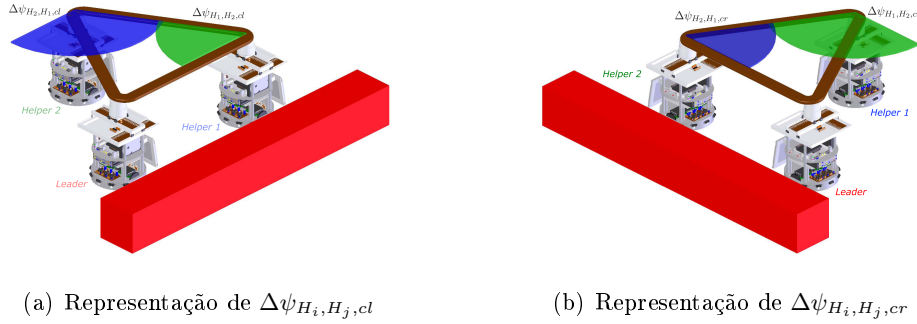


Figura 6.16: Representação dos ângulos característicos para o robô  $H_i$  relativamente ao robô  $H_j$  para o comportamento *Column*, isto é, de  $\Delta\psi_{H_i,H_j,c}$ , com  $i = \{1, 2\}$  e  $j = \{1, 2\} \wedge j \neq i$ .

Os valores de  $R_{H_i,L}$  e  $R_{H_i,H_j}$  para o comportamento *Column*, isto é,  $R_{H_i,L,c}$  e  $R_{H_i,H_j,c}$  são dados por:

$$R_{H_i,L,c} = \begin{cases} +1 & , \text{ se } i = 1 \\ -1 & , \text{ se } i = 2 \end{cases} \quad (6.46)$$

$$R_{H_i, H_j, c} = \begin{cases} +1 & , \text{ se } i = 1 \wedge j = 2 \\ -1 & , \text{ se } i = 2 \wedge j = 1 \end{cases} \quad (6.47)$$

As variáveis  $\alpha_{H_i, cl}$  e  $\alpha_{H_i, cr}$ , com  $i = \{1, 2\}$ , são de sinalização ao comportamento *Column* e correspondem à passagem paralela junto ao obstáculo detectado do lado esquerdo e à passagem paralela junto ao obstáculo detectado do lado direito, respectivamente.

$$\alpha_{H_i, cl} = \begin{cases} 1 & , \text{ se } U_{H_1, obs}(\phi_{H_1}) > 0 \wedge U_{H_2, obs}(\phi_{H_2}) \leq 0 \wedge \\ & |\psi_{H_1, L} - \frac{\pi}{2}| \leq \theta_3 \\ 0 & , \text{ caso contrário} \end{cases} , \quad i = \{1, 2\} \quad (6.48)$$

$$\alpha_{H_i, cr} = \begin{cases} 1 & , \text{ se } U_{H_1, obs}(\phi_{H_1}) \leq 0 \wedge U_{H_2, obs}(\phi_{H_2}) > 0 \wedge \\ & |\psi_{H_2, L} - \frac{\pi}{2}| \leq \theta_3 \\ 0 & , \text{ caso contrário} \end{cases} , \quad i = \{1, 2\} \quad (6.49)$$

em que  $U_{H_i, obs}$  é a função potencial da dinâmica referente ao campo vectorial da dinâmica dos obstáculos detectados pelo robô  $H_i$  (*Helper 1* ou *Helper 2*) dada de igual forma pela equação 5.19.  $f_{H_i, obs}$  é o campo vectorial resultante da dinâmica dos obstáculos detectados pelo robô  $H_i$  (*Helper 1* ou *Helper 2*) dada pela equação 5.10.  $\theta_3$ , como já foi referido anteriormente, é um parâmetro que foi definido como sendo  $\frac{\pi}{18} \text{ rad}$ .  $\psi_{H_i, L}$  representa o ângulo segundo o qual o robô  $H_i$  (*Helper 1* ou *Helper 2*) “vê” o robô *Leader*.

Caso se verifique um valor positivo das equações 6.48 ou 6.49 então o comportamento *Column* é activado, como se pode ver pela equação:

$$\gamma_{H_i,c} = \begin{cases} 1 & , \text{ se } \alpha_{H_i,cl} = 1 \vee \alpha_{H_i,cr} = 1 \\ 0 & , \text{ caso contrário} \end{cases}, \quad i = \{1, 2\} \quad (6.50)$$

## 6.2 Controlo de velocidade

A velocidade linear de cada robô *Helper*, é controlada por forma que em todos os instantes o respectivo robô tente anular o deslocamento  $\Delta d_{H_{any}}$  (equação 6.7) isto é, fazendo com que a junta rotacional do respectivo suporte do objecto vá para o centro do próprio robô *Helper*.

O sistema dinâmico que controla a velocidade de cada robô *Helper* (equação 6.2) erige um atractor para a velocidade desejada,  $V_{desired,H_{any}}$ , em que este valor é dado por:

$$V_{desired,H_{any}} = \begin{cases} +\vartheta_L - \frac{\Delta d_{H_{any},L}}{k_{v_{H_{any}}}} & , \text{ se } \Delta d_{H_{any},L} < 0 \\ -\vartheta_L - \frac{\Delta d_{H_{any},L}}{k_{v_{H_{any}}}} & , \text{ caso contrário} \end{cases} \quad (6.51)$$

onde  $k_{v_{H_{any}}}$  representa um parâmetro (ver equação 6.55) que define o aumento da velocidade para um dos robôs *Helpers* dependendo do lado para o qual a equipa tem que curvar e  $\Delta d_{H_{any},L}$  representa a medida que indica se o robô  $H_{any}$  está próximo ou afastado da posição desejada relativamente ao robô *Leader* sendo este valor negativo ou positivo, respectivamente.

$$\Delta d_{H_{any},L} = D - \sqrt{[x_{s,H_{any}} + D \times \cos(\theta_{H,L})]^2 + [y_{s,H_{any}} + D \times \sin(\theta_{H,L})]^2} \quad (6.52)$$

sendo

$$\theta_{H,L} = \theta_{e,H_{any}} + k_{\theta_{H_{any}}} \quad (6.53)$$

em que  $(x_{s,H_{any}}, y_{s,H_{any}})$  representam as coordenadas em que o apoio do objecto se encontra no suporte do robô  $H_{any}$  e  $D$  representa o comprimento do objecto a transportar entre o robô  $H_{any}$  e o *Leader*.  $\theta_{e,H_{any}}$  representa o valor medido pela junta rotacional do robô  $H_{any}$  (ver figuras 6.1 e 6.7) e  $k_{\theta_{H_{any}}}$  representa o ângulo que o *Leader* está desfazado em relação ao ângulo em que o *Helper* “vê” o grupo, e que é dado por:

$$k_{\theta_{H_{any}}} = \begin{cases} 0 \text{ rad} & , \quad \text{se } n = 2 \\ +\frac{\pi}{6} \text{ rad} & , \quad \text{se } n = 3 \wedge any = 1 \\ -\frac{\pi}{6} \text{ rad} & , \quad \text{se } n = 3 \wedge any = 2 \end{cases} \quad (6.54)$$

em que  $n$  representa o número de robôs que constitui a equipa de transporte.

É de salientar que o robô *Leader* comunica a sua velocidade linear e o valor do ângulo em que o grupo se encontra relativamente a ele próprio.

$$k_{v_{H_{any}}} = \begin{cases} k_0 & , \quad \text{se } n = 2 \\ k_1 & , \quad \text{se } n = 3 \wedge [(any = 1 \wedge \alpha_{H_{any},tl} = 1) \vee (any = 2 \wedge \alpha_{H_{any},tr} = 1)] \\ k_2 & , \quad \text{se } n = 3 \wedge \overline{[(any = 1 \wedge \alpha_{H_{any},tl} = 1) \vee (any = 2 \wedge \alpha_{H_{any},tr} = 1)]} \end{cases} \quad (6.55)$$

em que  $n$  representa o número de robôs que constitui a equipa de transporte.

Os valores que apresentaram melhor comportamento nas curvaturas foram:  $k_0 = 1/10$ ,  $k_1 = 1/6$  e  $k_2 = 1/2$ . Estes valores foram escolhido por simulação (sofrendo um pequeno ajuste aquando da implementação) tendo em conta que caso a equipa constituída por três robôs tenha que curvar para a esquerda o robô *Helper 1* tenha uma velocidade superior à do robô *Helper 2*, e caso tenha que curvar para a direita o robô *Helper 2* tenha uma velocidade superior à do robô *Helper 1*.

Discretizando a equação 6.2 pelo método de *Euler progressivo*, tem-se:

$$\vartheta_{H_{any}}(t_k) = \vartheta_{H_{any}}(t_{k-1}) - \lambda_{v_{H_{any}}} \times [\vartheta_{H_{any}}(t_{k-1}) - V_{desired, H_{any}}(t_k)] \times dt, \quad \forall k \in N^+ \quad (6.56)$$

### 6.3 Sumário

Para qualquer uma das equipas, as variáveis comportamentais para cada robô são a direcção de navegação e a velocidade de translação.

Independentemente do número de robôs, são necessários no máximo três comportamentos, para que assim os robôs:

- se movimentem em frente ou recuem (*Forward/Backward*);
- se desviem de obstáculos, à esquerda ou à direita (*Turn Left* ou *Turn Right*);
- se movimentem paralelamente a um obstáculo (*Column Left* ou *Column Right*).

Relativamente à configuração geométrica, caso se deseje que os robôs naveguem numa configuração que não seja a referida neste capítulo, apenas é necessário alterar os parâmetros  $\Delta\psi_{H_i, L, behaviour}$  e  $\Delta\psi_{H_i, H_j, behaviour}$  da equação 6.22 e 6.23, respectivamente.





## Capítulo 7

# Resultados

---

Neste capítulo são apresentados resultados da implementação da arquitectura de controlo distribuída, apresentada no capítulo anterior, em equipas de dois e de três robôs móveis autónomos. Para cada equipa são apresentados dois cenários da mesma implementação.

---

### 7.1 Equipa de dois robôs

Nas subsecções que se seguem, são apresentados dois cenários onde o desempenho dos robôs, *Leader* e *Helper*, para a equipa de dois robôs móveis autónomos, foi testado. Nos cenários apresentados, o robô *Leader*, robô com uma faixa vermelha, apenas possui o comportamento *evitar colisões com obstáculos* pelo facto dos actuadores estarem em malha aberta. O cálculo da posição do robô *Leader* é efectuado por hometria, o que faz com que esse cálculo, em malha aberta, acumule erro rapidamente. Este problema poderia ser resolvido utilizando uma câmara para a localização do alvo. Uma vez que o objectivo desta dissertação é implementar e validar em termos práticos a arquitectura de controlo distribuído desenvolvidas por Soares (2007), decidiu-se optar pela não inclusão do comportamento *dirigir-se para o alvo*. Assim, a dinâmica comportamental resultante do robô *Leader* é dada apenas pela contribuição da dinâmica *evitar colisões*

com obstáculos, movimentando-se, portanto, o robô *Leader* à “deriva”, o que não invalida o teste do desempenho do robô *Helper*.

### 7.1.1 Cenário 1 - O *Leader* entra numa passagem sem saída

Ambos os robôs partem segundo uma orientação de  $\frac{\pi}{2}$  rad. O comportamento por defeito do robô *Helper* é o comportamento *Forward/Backward*. Estas condições podem ser vistas no instante 01 da figura 7.1.

Do instante 02 para o 03 nota-se uma diminuição da direcção de navegação do robô *Helper* (robô com uma faixa azul) devido ao objecto não estar no centro do suporte quando iniciam a tarefa. Esta diminuição da direcção de navegação do robô *Helper* acontece para o mesmo levar o objecto para o seu centro. Nestes instantes o comportamento que está activo é o *Forward/Backward*. No instante 06 o robô *Helper* já detectou os obstáculos que se encontram à sua frente, e o comportamento *Turn* é activado para fazer com que o robô *Helper* mude a sua direcção de navegação de modo a evitar colidir com os obstáculos.

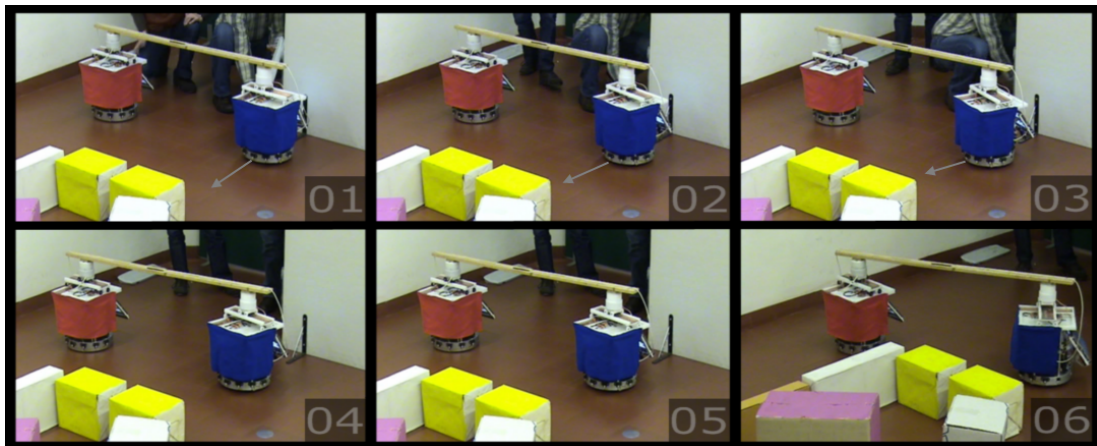


Figura 7.1: Implementação da arquitectura de controlo, aplicada à equipa de dois robôs móveis autónomos, num cenário bastante congestionado. Esta figura é constituída por 48 instantes retirados de um vídeo realizado aquando a implementação da arquitectura de controlo.

No instante 07 o robô *Helper* permanece sob a influência de obstáculos e “vê” o robô *Leader* numa direcção muito próxima da sua direcção de navegação, alterando o com-

portamento para o *Column*, mantendo-se este comportamento activo até ao instante 09. A partir do instante 10 a magnitude da contribuição *evitar colisões com obstáculos* do robô *Helper* é fraca, o que faz com que este tente colocar-se no comportamento *Forward/Backward*, mas como ao voltar-se para a esquerda detecta obstáculos, logo o comportamento *Turn* fica activo, o que faz com que oscile entre comportamentos até ao instante 12.

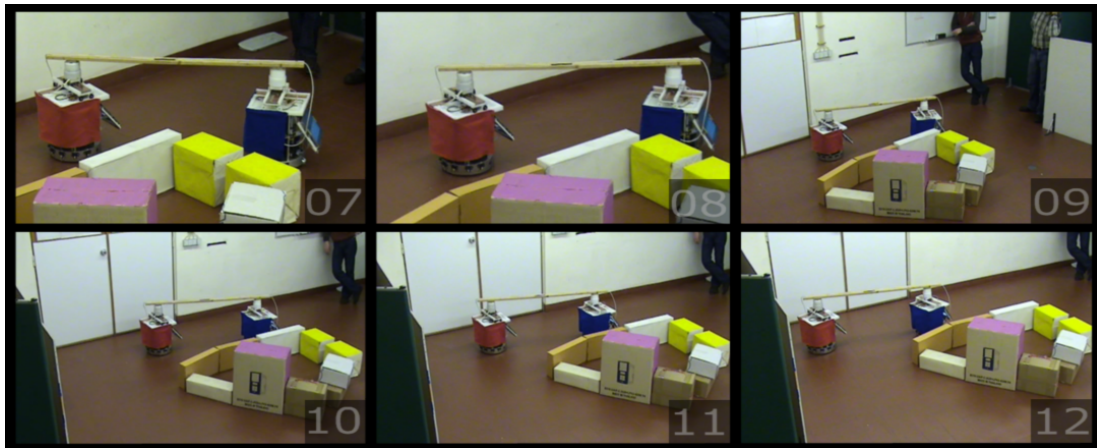


Figura 7.1 (cont. 1)

Entre os instantes 13 e 15, o robô *Helper* navega sob o controlo do comportamento *Column*. Entre os instantes 16 e 20 o robô *Helper* alterna consecutivamente entre os comportamentos *Turn* e *Forward/Backward*. Esta alternância deve-se ao facto da força repulsiva sentida pelo robô *Helper* ser fraca, o que faz com que este navegue sob o controlo do comportamento *Forward/Backward*, mas ao tentar colocar-se lado a lado com o robô *Leader* sente a força repulsiva maior, logo o comportamento *Turn* é activado, e assim sucessivamente.

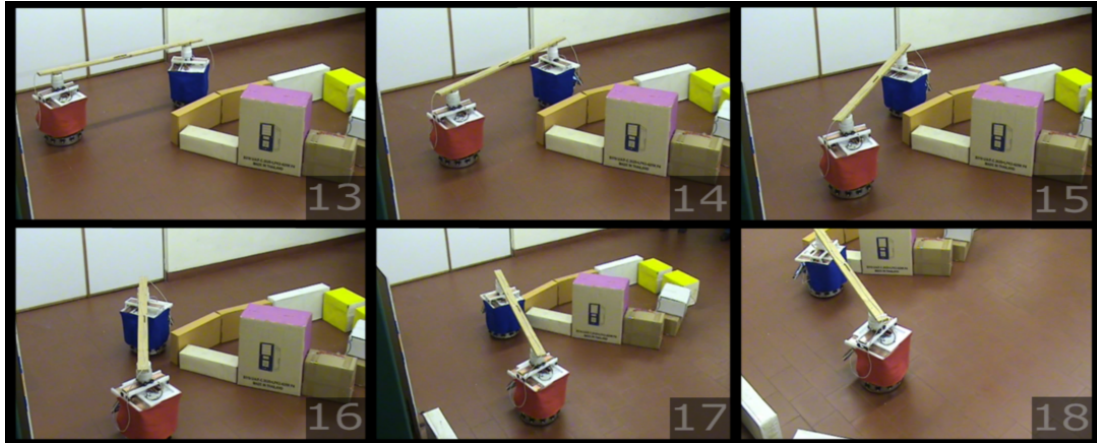


Figura 7.1 (cont. 2)

A partir do instante 20 até aos instante 25 o robô *Helper* navega no comportamento *Forward/Backward*, isto é, vai lado a lado com o robô *Leader*.

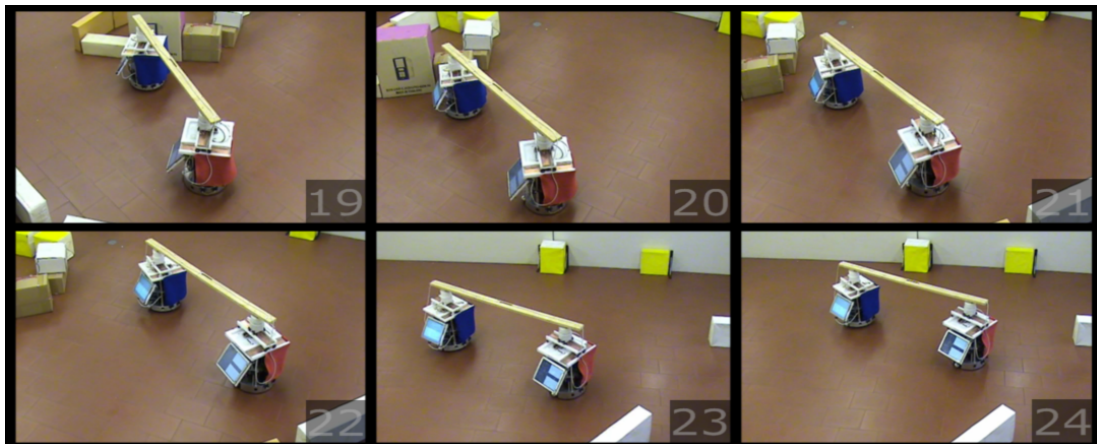


Figura 7.1 (cont. 3)

No instante 26 o robô *Helper* detecta um obstáculo e entra no comportamento *Turn*, para evitar colidir com o obstáculo. No instante seguinte, como este robô “vê” o *Leader* aproximadamente na sua direcção de navegação, o *Helper* passa a navegar sob o controlo do comportamento *Column*. Do instante 30 ao 32 através da informação comunicada, o robô *Helper* sabe que o *Leader* navega na sua direcção de navegação, o que faz com que o comportamento que governa a direcção de navegação do robô *Helper* seja o *Forward/Backward*, sendo a componente *Backward* a activa neste comportamento.

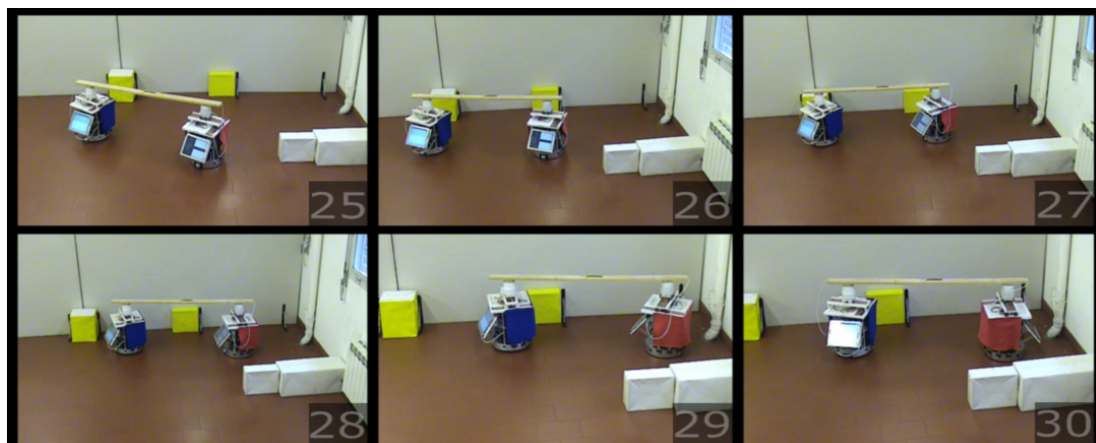


Figura 7.1 (cont. 4)

Do instante 33 ao 38 a direcção de navegação do robô *Helper* continua a ser governada pelo comportamento *ForwardBackward*, sendo que nestes instantes a componente *Forward* está activa neste comportamento.

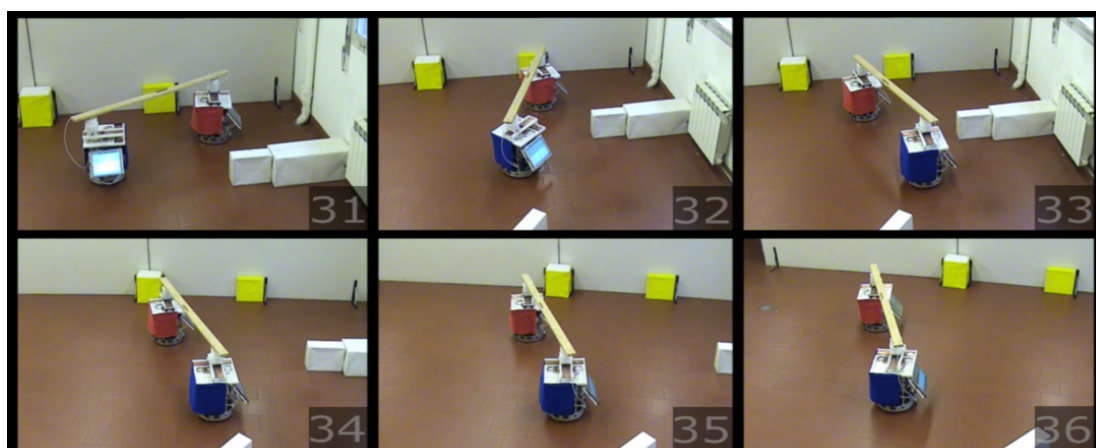


Figura 7.1 (cont. 5)

No instante 39 o robô *Helper* detecta os obstáculos e entra com comportamento *Turn*. No instante seguinte como este robô “vê” aproximadamente na sua direcção de navegação o robô *Leader*, navegando o robô *Helper* sob a regência do comportamento *Column*.



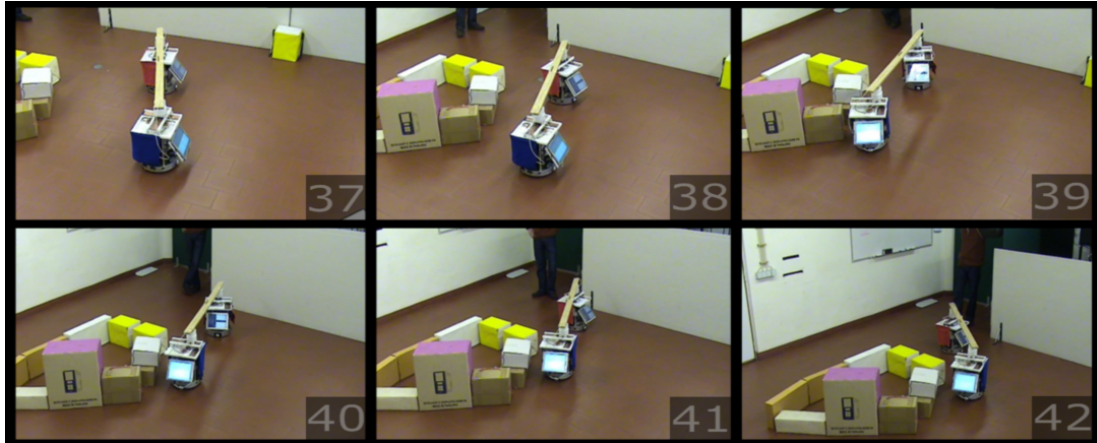


Figura 7.1 (cont. 6)

No instante 45 o robô *Helper* por meio da informação comunicada, sabe que a sua direcção de navegação está a ir de encontro com o percurso que o *Leader* está a tomar, devido a essa informação, o *Helper* a recua para facilitar a passagem ao robô *Leader*. No instante 46 o *Helper* detecta os obstáculos e entra no comportamento *Turn*. Nos instantes seguintes a direcção de navegação do robô *Helper* é governada pelo comportamento *Column*.

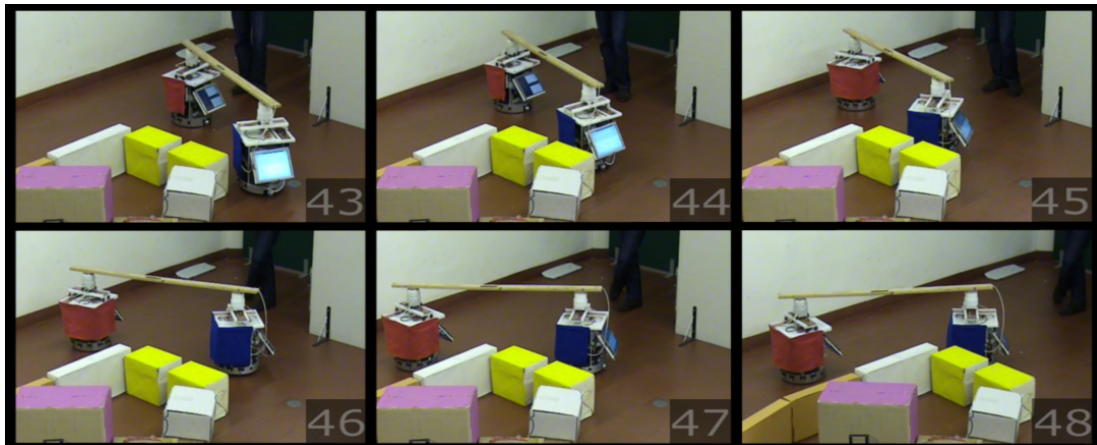


Figura 7.1 (cont. 7)

### 7.1.2 Cenário 2 - Passagens estreitas e passagem pela porta

Neste cenário, também ambos os robôs partem segundo uma orientação de  $\frac{\pi}{2}$  rad. O comportamento por defeito do robô *Helper* é o comportamento *Forward/Backward*.

Estas condições podem ser vistas no instante 01 da figura 7.2.

Nos instantes 01 e 02, o robô *Helper* navega no comportamento *Forward/Backward*. No instante 03 o *Helper* detecta os obstáculos à sua frente transitando para o comportamento *Turn*, até que no instante 04 “vê” o robô *Leader* na sua direcção de navegação e o comportamento *Column* fica activo.



Figura 7.2: Implementação da arquitectura de controlo, aplicada à equipa de dois robôs móveis autónomos, num cenário bastante congestionado.

Entre o instante 09 e o 11 o robô *Helper* oscila entre o comportamento *Turn* e o *Forward/Backward*. A partir do instante 12 a direcção de navegação do robô *Helper* passa a ser governada pelo comportamento *Column*.

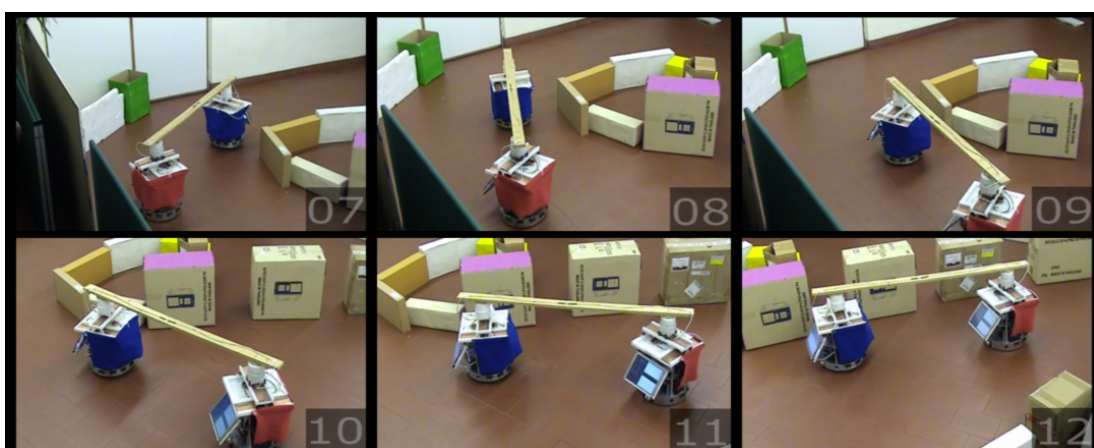


Figura 7.2 (cont. 1)

Entre os instantes 13 e 23 o robô *Helper* oscila entre o comportamento *Turn* e o

*Forward/Backward.*

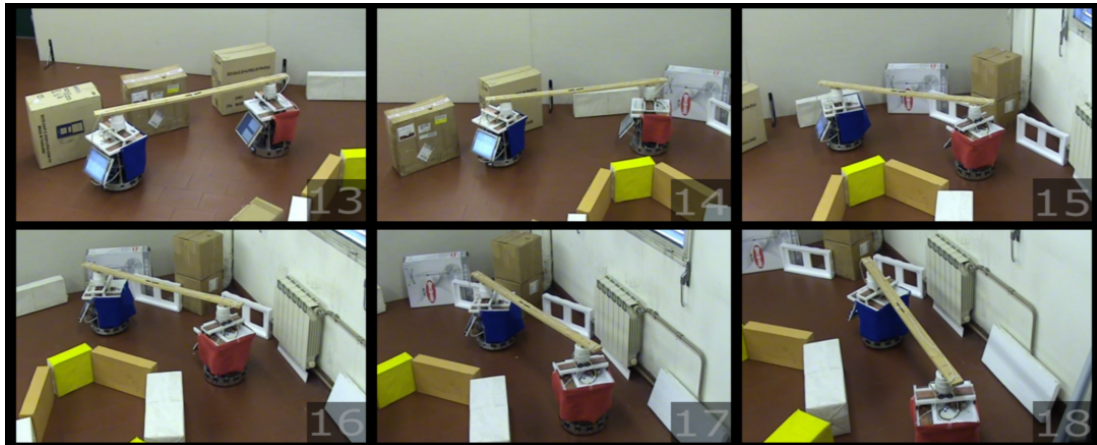


Figura 7.2 (cont. 2)

Entre os instantes *24* e *31* o comportamento que está activo é o *Column*.



Figura 7.2 (cont. 3)





Figura 7.2 (cont. 4)

A partir do instante 31 um humano abre a porta para a equipa de robôs sair do laboratório. A partir do instante 32 até ao instante 35 o robô *Helper* detecta mais espaço do seu lado esquerdo e fica novamente em transições consecutivas entre o comportamento *Turn* e o *Forward/Backward*. Por fim, o robô *Helper* detecta a porta e navega sob a regência do comportamento *Column* até atravessar por completo a passagem da porta.



Figura 7.2 (cont. 5)

Na figura 7.3, estão representados os atractores para a dinâmica da direcção de navegação do robô *Helper* neste cenário. Nesta figura pode-se verificar que em todos os momentos a direcção de navegação do robô *Helper* acompanha (ou tenta acompanhar, no caso de oscilações no comportamento) o atractor especificado para a sua direcção de navegação. É de salientar que existe a acumulação de bastante erro na actualização

da sua direcção de navegação. Esta acumulação do erro também se dá no cálculo da direcção que especifica o atractor para a direcção de navegação. Como no cálculo da dinâmica resultante é efectuada a diferença entre a direcção de navegação e a direcção desejada, então o erro acumulado anula-se.

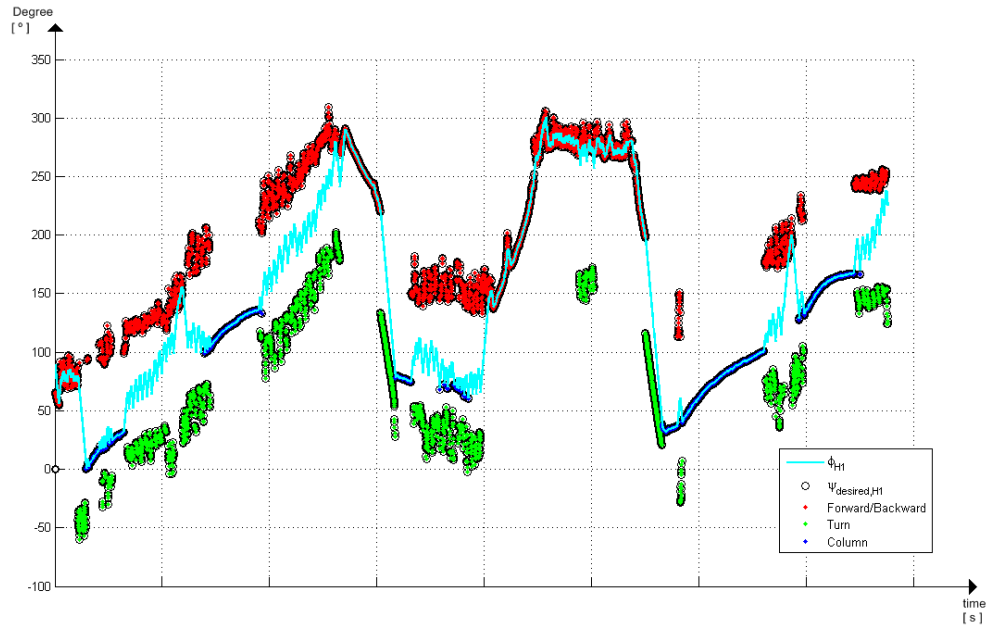


Figura 7.3: Representação gráfica do atractor desejado para a direcção de navegação do robô *Helper* e a sua direcção de navegação, referente a um cenário implementado para a equipa de dois robôs móveis autónomos. As circunferências a preto representam a direcção do atractor desejado ao longo para cada iteração do ciclo computacional, e a cor que as preenche representa o comportamento activo. A linha a azul claro representa a direcção de navegação do robô *Helper*.

## 7.2 Equipa de três robôs

Nas subsecções que se seguem, são também apresentados dois cenários onde o desempenho dos robôs, *Leader*, *Helper 1* e *Helper 2*, da equipa de três robôs móveis autónomos, foi testado. Nos cenários apresentados, o robô *Leader*, robô com uma faixa vermelha, tal como nos cenários apresentados para a equipa de dois robôs móveis autónomos, apenas possui o comportamento *evitar colisões com obstáculos*. A não inclusão do comportamento *dirigir-se para o alvo* deve-se ao mesmo motivo que o da equipa de dois robôs. Tal como na equipa de dois robôs, uma vez que o objectivo desta dissertação é implementar e validar em termos práticos a arquitectura desenvolvida por Soares (2007), decidiu-se optar pela não inclusão do comportamento *dirigir-se para o alvo* no robô *Leader*. Assim, a dinâmica comportamental resultante do robô *Leader* é dada apenas pela contribuição dos obstáculos, movimentando-se, portanto, o robô *Leader* à “deriva”, o que não invalida o teste do desempenho dos robôs *Helpers*.

### 7.2.1 Cenário 1 - Humanos colocam carga nos robôs

Os robôs iniciam a sua tarefa próximos de um obstáculo, instantes de 01 a 03, o que faz com que a equipa navegue sob a regência do comportamento *Column*. A partir do instante 04 os *Helpers* navegam sob o *Forward/Backward*, sendo a componente *Forward* a activa. No instante 05 dois humanos vão de encontro à equipa de robôs e colocam-lhe uma carga para transportarem. No instante seguinte o robô *Leader* detecta a obstrução provocada pela presença dos humanos e desvia-se, continuando sob o comportamento *Forward/Backward*, mas como o *Leader* navega numa direcção que vai de encontro à direcção do robô *Helper 2*, a componente activa deste comportamento é a *Backward*.

Até ao instante 08 continuam os robôs *Helpers* continuam com a componente *Backward* activa do comportamento *Forward/Backward*, e a partir desse instante até ao instante 13 passa a estar activa a componente *Forward* deste mesmo comportamento, uma vez que a direcção de navegação do *Leader* deixa de estar em “conflito” com a direcção de navegação do *Helper* referido.

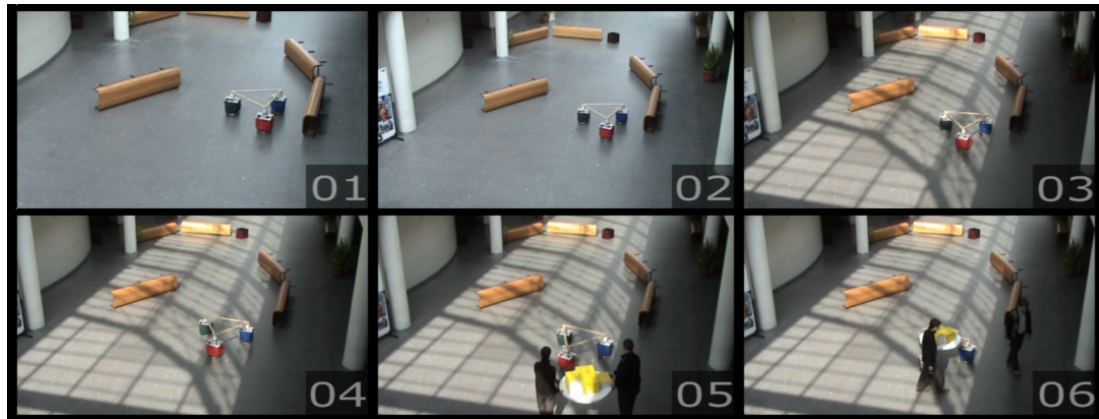


Figura 7.4: Equipa de três robôs - Cenário 1

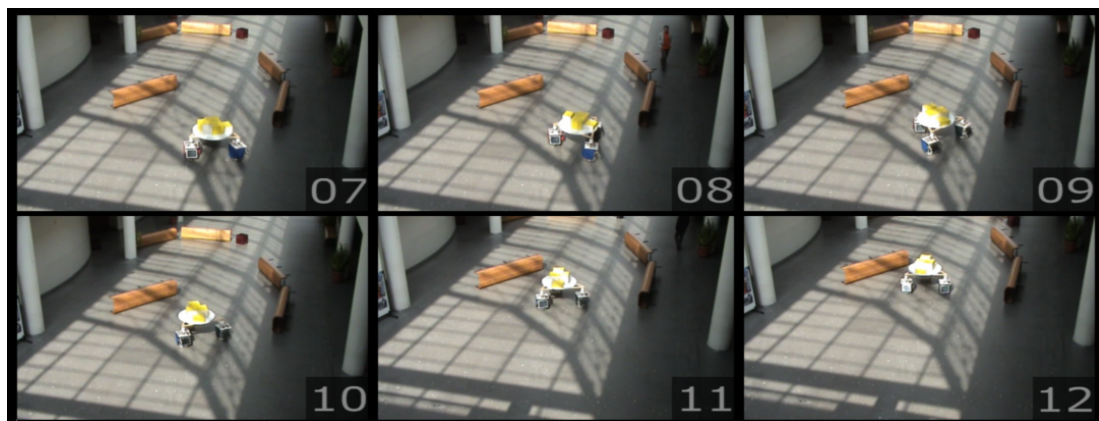


Figura 7.4 (cont. 1)

Entre os instantes *14* e *15* a equipa desvia-se do obstáculo que está do lado esquerdo da mesma, sob a regência do comportamento *Turn* com a componente *Right* deste comportamento activa. Quando a equipa têm o caminho livre de obstruções, volta ao comportamento por defeito, isto é, ao comportamento *Forward/Backward*, isto entre os instantes *16* a *19*.

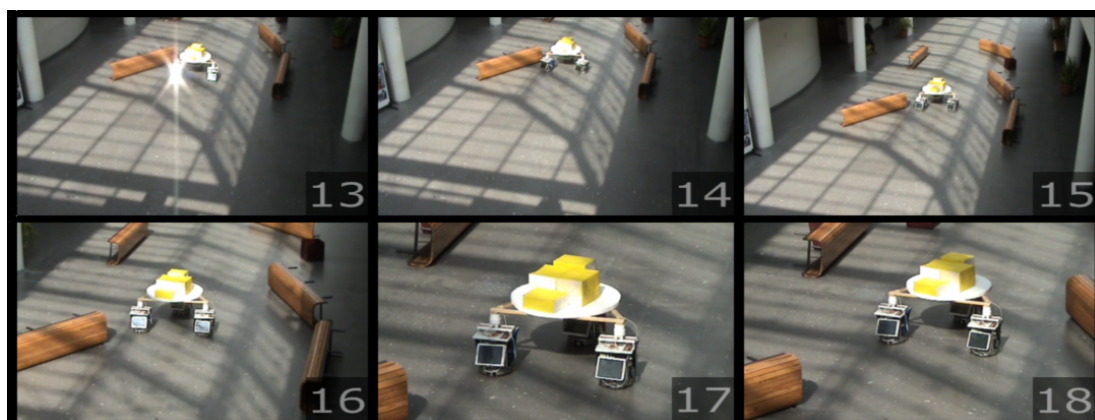


Figura 7.4 (cont. 2)

Entre os instantes *20* e *22* a equipa desvia-se do obstáculo que está do lado direito, sob a governação do comportamento *Turn*, sendo a componente activa a *Left* deste mesmo comportamento.



Figura 7.4 (cont. 3)

Do instante *23* ao *26* a equipa navega paralelamente ao obstáculo que está do lado direito da mesma, e caso os humanos não impedissem a passagem da equipa, esta seguiria em frente. Assim a presença dos humanos fez com que a equipa voltasse a curvar. Entre os instantes *27* a *30*, como a direcção de navegação do *Leader* aponta novamente numa direcção que vai de encontro com a direcção de navegação de uma dos *Helpers*, desta vez com a direcção de navegação do *Helper 1*, os *Helpers* navegam sob o comportamento *Forward/Backward* sendo a componente *Backward* a activa.





Figura 7.4 (cont. 4)

A partir do instante *31* até ao final deste cenário, a equipa continua a navegar sob o controlo do comportamento *Forward/Bakward*, sendo agora a componente *Forward* a activa.

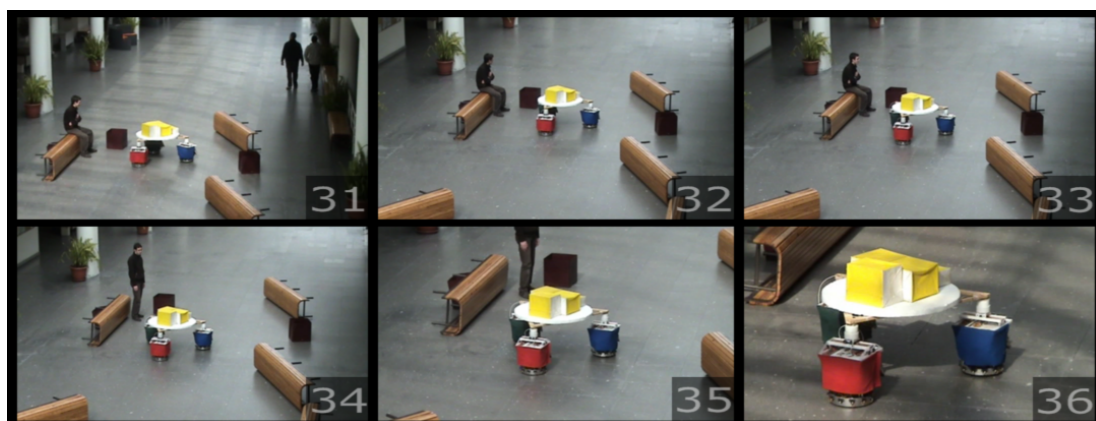


Figura 7.4 (cont. 5)

### 7.2.2 Cenário 2 - Equipa desvia-se de um obstáculo móvel

Neste cenário, a mesma equipa de robôs móveis foi colocada frente a frente com um humano a empurrar uma carga de grande dimensão. Do instante *01* ao *05* como não existem obstruções os robôs *Helpers* navegam sob o domínio do comportamento *Forward/Backward*. No instante *06* o *Leader* detecta a obstrução da carga empurrada pelo humano.

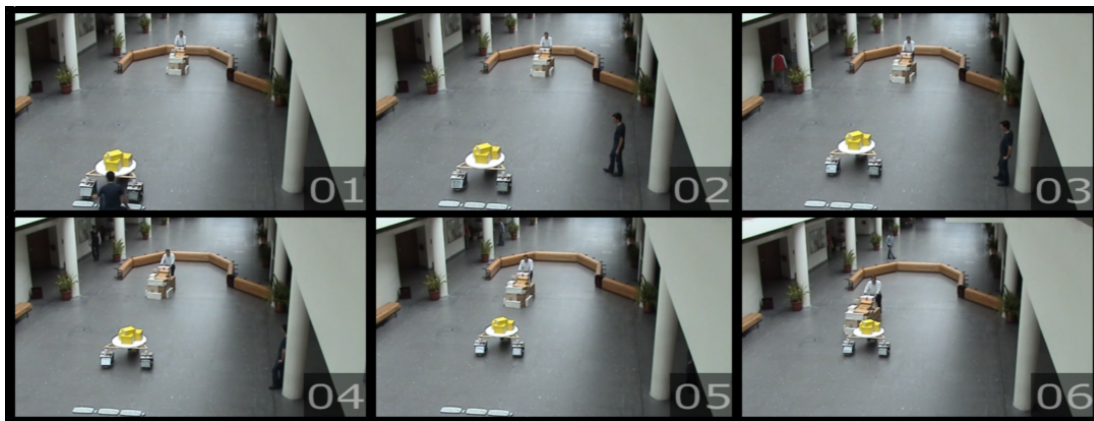


Figura 7.5: Equipa de três robôs - Cenário 2

Nos instantes *07* e *08* o robô *Helper 1* como detecta do seu lado esquerdo a carga transportada pelo humano, é activado o comportamento *Turn*, sendo a componente *Right* deste a responsável pelo desvio da equipa. Do instante *09* ao instante *12* os *Helpers* navegam sob a regência da componente *Forward* do comportamento *Forward/Backward*.



Figura 7.5 (cont. 1)

Nos instantes *13* e *14* como a direcção de navegação do robô *Leader* vai de encontro à direcção de navegação do robô *Helper 1*, os *Helpers* activam a componente *Backward* do comportamento *Forward/Backward*. A partir do instante *15* a direcção do robô *Leader* deixa de estar a apontar numa direcção que vá de encontro com a de um *Helper* e então os *Helpers* deixam de navegar com a componente *Backward* do comportamento *Forward/Backward* a dominar a dinâmica de atractores para a direcção de navegação.

Desde o instante 15 até ao 18 dado que o robô *Helper 2* detecta uma obstrução do seu lado direito, o comportamento *Turn*, sendo a componente *Left*, do mesmo, a activa, é o dominante.

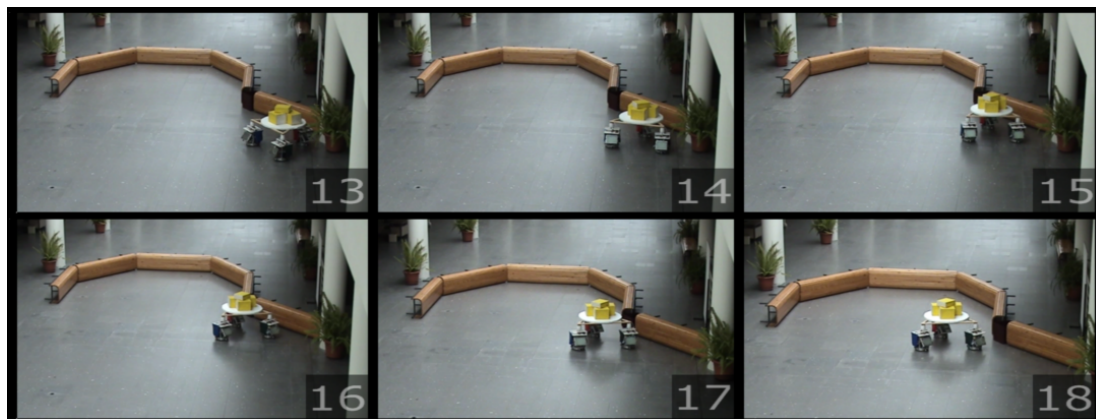


Figura 7.5 (cont. 2)

Do instante 19 ao 23 os robôs *Helpers* não sentem obstruções e o robô *Leader* navega em frente, o que faz com que o comportamento *Forward/Backward* o que domina a dinâmica dos atratores para a direcção de navegação dos *Helpers*. No instante 24 o robô *Leader* detecta obstruções no caminho que a equipa está e como estas são mais fortes do lado esquerdo, então este toma a decisão de virar à direita.

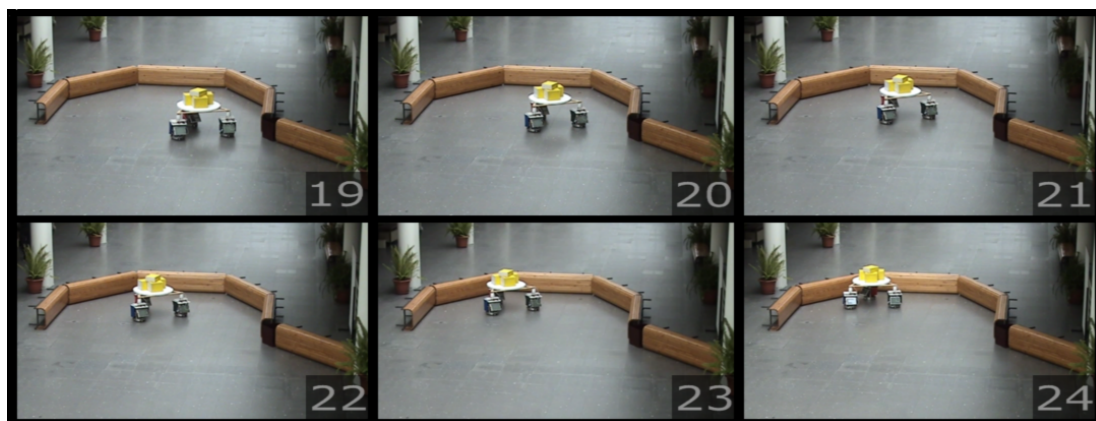


Figura 7.5 (cont. 3)

Como o robô *Leader* navega junto dos obstáculos, e como o *Helper 1* detecta obstruções do seu lado esquerdo, a equipa navega até ao instante 32 paralelamente ao obstáculo.



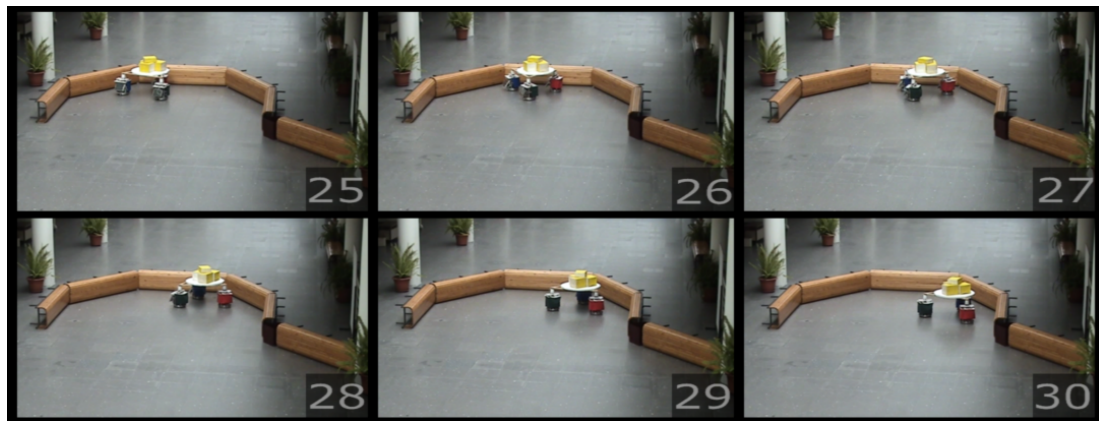


Figura 7.5 (cont. 4)

A partir do instante 33 a equipa tem o caminho totalmente livre, pelo que não é detectada nenhuma obstrução e o robô *Leader* navega em frente, o que faz com que a componente *Forward* do comportamento *Forward/Backward* domine a dinâmica referente à direcção de navegação dos robôs *Helpers*.



Figura 7.5 (cont. 5)

Na figura 7.6, estão representados os atractores de uma implementação dos robôs  $H_i$  (*Helper 1* ou *Helper 2*) (figura (a) e (b), respectivamente), de uma equipa de três robôs móveis autónomos. Nesta figura pode-se verificar que em todos os momentos a direcção de navegação dos robôs *Helpers* acompanha (ou tenta acompanhar, no caso de oscilações no comportamento) o atractor especificado para a sua direcção de navegação. É de salientar que existe a acumulação de bastante erro na actualização da sua direcção de navegação. Esta acumulação do erro também se dá no cálculo da direcção que especifica o atractor para a direcção de navegação. Como no cálculo da dinâmica resultante é efectuada a diferença entre a direcção de navegação e a direcção desejada, então o erro acumulado anula-se.

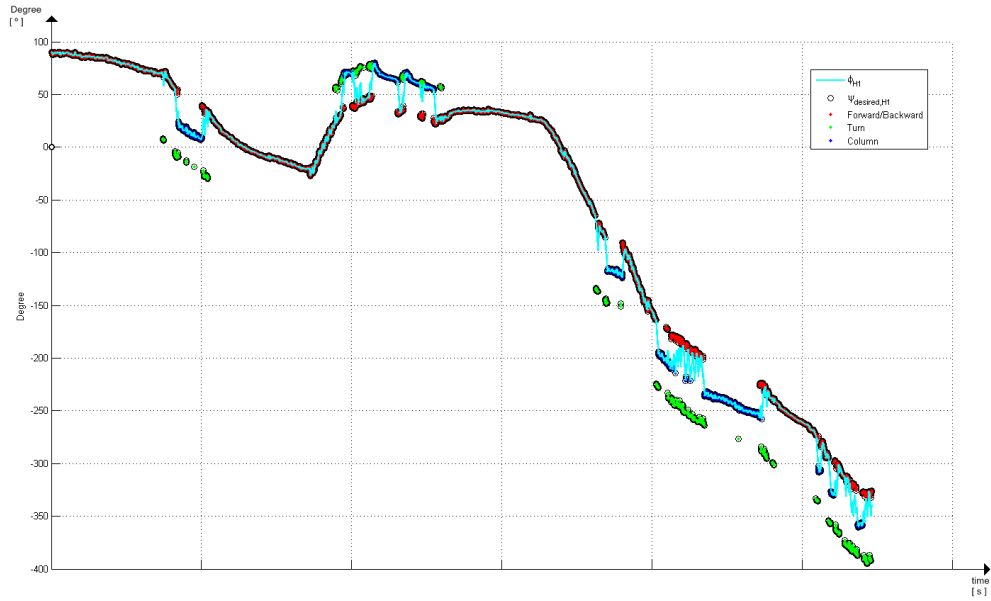
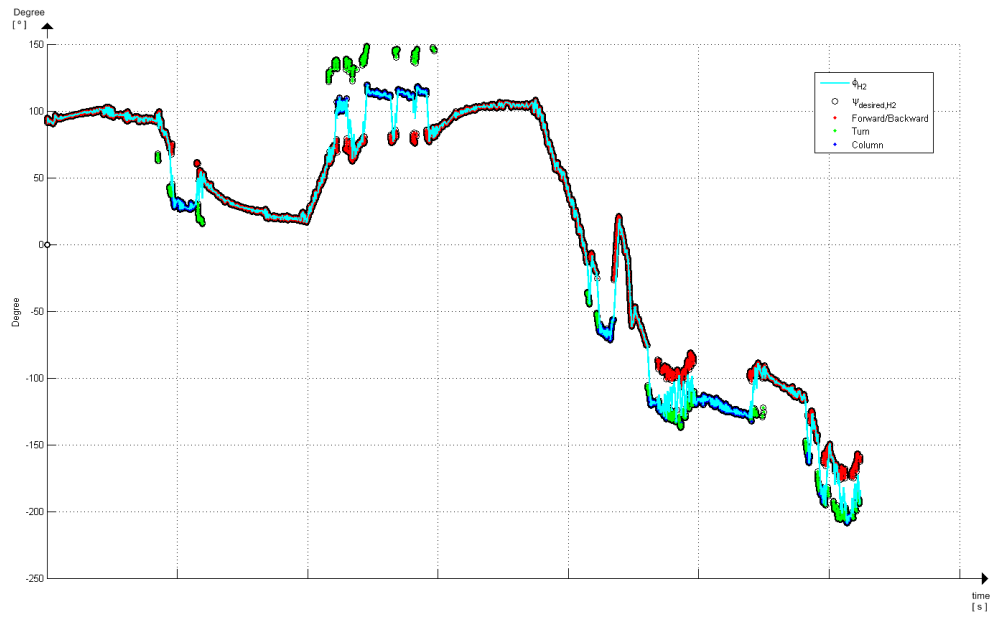
(a) *Helper 1* da equipa de três robôs(b) *Helper 2* da equipa de três robôs

Figura 7.6: Representação gráfica do atractor desejado para a direcção de navegação dos robôs *Helpers* e as respectivas direcções de navegação, referente a um cenário implementado para a equipa de três robôs móveis autónomos. As circunferências a preto representam a direcção desejada para cada iteração do ciclo computacional, e a cor que as preenche representa o comportamento activo. A linha a azul celeste representa a direcção de navegação do robô *Helper* que o gráfico representa.

### 7.3 Sumário

Foi demonstrado, que uma simples arquitectura de controlo baseada em atractores móveis de sistemas dinâmicos pode ser usada para controlar equipas de dois e de três robôs que em conjunto transportam um objecto de grandes dimensões num ambiente com obstáculos estáticos e/ou dinâmicos.

Importante a salientar, é o comportamento do(s) robô(s) *Helper(s)* que é estável e as trajectórias geradas são suaves.

A contribuição do alvo, no robô *Leader* foi desactivada, uma vez que a posição do alvo é determinada através da regra de hometria, que não permite obter bons resultados, dado o facto dos controladores estarem em malha aberta, isto é, para percursos relativamente longos o erro acumulado é grande, logo os robôs nunca têm uma informação correcta da sua posição.

De referir ainda que sempre que e quando for mais benéfico podem-se apenas utilizar dois comportamentos. Por exemplo se o ambiente que rodeia os robôs é um ambiente estreito pode não se justificar o comportamento *Forward/Backward*. Nestes casos utilizam-se apenas os comportamentos *Column* e *Turn* que, tal como se pode ver por algumas das implementações efectuadas e registadas em vídeo no anexo B, em nada impede os robôs de concluírem com sucesso a sua tarefa.



## Capítulo 8

# Conclusão

---

Neste capítulo é feito um resumo dos principais resultados desta dissertação. No final é ainda feita uma descrição de perspectivas quanto a trabalho futuro.

---

### 8.1 Síntese

A abordagem para a geração de comportamentos baseada em atractores de sistemas dinâmicos, foi usada nesta dissertação, como ferramenta teórica base, para a implementação da arquitectura de controlo distribuída para a coordenação de equipas de robôs móveis autónomos, que em conjunto devem transportar um objecto de grandes dimensões. São vários e complexos os problemas que existem neste tipo de trabalhos:

- i) manutenção de uma configuração geométrica fixa durante o movimento: um dos problemas principais é manutenção de uma configuração geométrica fixa durante o movimento, uma vez que, durante a tarefa de transporte é necessário que os robôs coordenem os seus movimentos de forma precisa e com uma margem de erro mínima, caso contrário o objecto transportado poderá cair;
- ii) ambientes desconhecidos *a priori*: o facto dos robôs não terem um conhecimento prévio do ambiente que os rodeia obriga a que a equipa seja flexível, isto é, que se

adapte a “qualquer” ambiente, pelo que as trajectórias têm de ser calculadas em tempo-real;

- iii) evitar colisões com obstáculos estáticos e/ou dinâmicos: o problema de transporte torna-se ainda mais agravado quando os robôs têm que evitar colisões com obstáculos, isto porque, os robôs em conjunto têm que se desviar dos obstáculos e simultaneamente coordenar os seus movimentos por forma a que o objecto transportado não caia;

Os resultados, documentados em vídeo, das implementações com equipas de dois e de três robôs móveis autónomos, mostram o bom desempenho das mesmas nos vários cenários testados (cenários com obstáculos estáticos e dinâmicos e ainda cenários bastante congestionados), embora a velocidade de translação do robô *Leader* nunca tenha excedido os 10 *cm/s*. Notou-se que um aumento desta velocidade causaria uma variação mais acentuada da direcção de navegação do(s) robô(s) *Helper(s)*, uma vez que tal aumento de velocidade causaria mais variações no suporte do objecto, dado que a compensação deste deslocamento é traduzida num ângulo que é incluído na direcção de navegação desejada para o respectivo robô.

O comportamento resultante é estável e as trajectórias geradas são suaves, porque à medida que a informação sensorial e/ou a informação comunicada varia, o campo vectorial do sistema dinâmico de cada robô *Helper* adapta-se e gera concomitantes soluções atractoras.

Através dos resultados obtidas pela implementação prática, verificou-se que a arquitectura de controlo distribuída (baseada em sistemas dinâmicos não lineares), desenvolvida por Soares (2007), é válida, para as equipas de dois e de três robôs móveis autónomos, mesmo com recursos bastante reduzidos, pelo que se espera obter ainda melhores resultados com recursos melhores do que os utilizados para a realização desta dissertação. Tendo em conta os resultados das implementações realizadas para equipas de dois e de três robôs móveis autónomos, espera-se de igual forma um bom desempenho em equipas constituídas por mais do que três robôs móveis autónomos, dado que a arquitectura de controlo apenas necessita de no máximo dois pontos de referência, isto para qualquer

que seja o número de robôs que constituem a equipa.

Como principal desvantagem a apontar a este trabalho é o facto dos robôs apenas poderem navegar em superfícies relativamente planas, dado o tipo de suporte do objecto aqui utilizado ser constituído por juntas passivas. Além desta restrição em termos de superfícies onde as equipas podem navegar, é de salientar que os robôs são meramente reactivos, isto é, não têm qualquer tipo de inteligência, uma vez que a arquitectura não integra processos cognitivos, tais como a percepção, antecipação e aprendizagem, para garantir com maior rigor e facilidade o sucesso da equipa.

## 8.2 Trabalho futuro

Como trabalho futuro pretende-se desenvolver um sistema autónomo para a monitorização e sintonização automática dos parâmetros que por enquanto são escolhidos por estudo analítico e simulação, embora esses parâmetros sofram ajustes aquando da passagem da simulação à prática. O benefício imediato, é permitir a generalização da utilização destes sistemas a potenciais utilizadores sem conhecimentos técnicos específicos.

É igualmente um objectivo, a extensão das arquitecturas por forma a permitir o transporte cooperativo por equipas constituídas por humanos e robôs. A realização deste objectivo implica a eliminação da necessidade de comunicação explícita entre os robôs, o que é um desafio considerável. Pretende-se também efectuar transporte cooperativo de objectos com equipas de robôs móveis autónomos equipados com manipuladores. Isto implica a coordenação de cada plataforma móvel em relação ao seu manipulador e vice-versa e ainda a coordenação entre as várias plataformas que constituem a equipa (figura 8.1).

Um outro objectivo é dotar os robôs com capacidades cognitivas (exemplo: memória, antecipação). Neste sentido e no âmbito do Projecto Europeu JAST - *Joint Action Science Technology* (FP6-IST2, Proj. n.o 003747) estão a ser realizadas experiências em



Figura 8.1: Ilustração de parte do trabalho que se pretende realizar a curto/médio prazo, no trabalho de doutoramento que me propus realizar.

colaboração com dois dos parceiros (*Nijmegen Institute for Cognition and Information* e *Max Planck Institute for Human Cognitive and Brain Sciences*). Estes têm como objectivo investigar como é que cada agente humano desenvolve uma estratégia de antecipação e adaptação aos companheiros de equipa. A ideia é depois implementar estas estratégias nas arquitecturas dos robôs móveis autónomos reais, como ilustra a figura 8.1, no âmbito do meu programa de trabalhos de doutoramento, para o qual me foi atribuída uma bolsa (ref.: *SFRH/BD/38885/2007*), pela F.C.T. (Fundação para a Ciência e Tecnologia).



## Apêndice A

# Valores dos parâmetros

### A.1 Capítulo 3

$R_{wheel} = 4,5 \text{ cm}$  - Raio das rodas motrizes.

$d_{wheel} = 26,5 \text{ cm}$  - Distância entre as rodas motrizes.

### A.2 Capítulo 5

$\Delta\theta = \frac{\pi}{6}$  - Espaçamento angular entre cada par consecutivo de sensores.

$\lambda_{tar} = 2$  - Magnitude da dinâmica *dirigir-se para o alvo*.

$\beta_1 = 40$  - Magnitude da *evitar colisões com obstáculos*.

$\beta_2 = 25$  - Taxa de decaimento.

$\lambda_v = 30$  - Magnitude de atracção para a velocidade linear.

### A.3 Capítulo 6

$\alpha_{H_{any}} = 1$  - Factor de ganho do deslocamento do objecto.

$\theta_3 = \frac{\pi}{18}$  - Gama angular do comportamento *Column*.

### A.3.1 Equipa de dois robôs

- $\theta_1 = \frac{3\pi}{4}$  - Limite inferior para activação da variável de sinalização  
da componente *Forward* do comportamento *Forward/Backward*.
- $\theta_2 = \frac{5\pi}{4}$  - Limite superior para activação da variável de sinalização  
da componente *Forward* do comportamento *Forward/Backward*.

### A.3.2 Equipa de três robôs

- $\theta_1 = \frac{\pi}{4}$  - Limite inferior para activação da variável de sinalização  
da componente *Forward* do comportamento *Forward/Backward*.
- $\theta_2 = \frac{3\pi}{4}$  - Limite superior para activação da variável de sinalização  
da componente *Forward* do comportamento *Forward/Backward*.

## Apêndice B

### Versão digital

+



# Bibliografia

- Ahmadabadi, M. and E. Nakano (2001). A constrain and move approach to distributed object manipulation. *in Proc. IEEE Transactions on Robotics and Automation* **17**, 157–172.
- Aiyama, Y., M. Hara, T. Yabuki, J. Ota and T. Arai (1999). Cooperative transportation by two four-legged robots with implicit communication. *Robotics and Autonomous Systems* **29**, 13–19.
- Bicho, E. (2000). *Dynamic approach to Behavior-Based Robotics: design, specification, analysis, simulation and implementation*. Shaker Vergal.
- Bicho, E. and G. Schöner (1997). The dynamic approach to autonomous robotics demonstrated on a low-level vehicle platform. *Robotics and Autonomous Systems* **21**, 23–25.
- Bicho, E., L. Louro and W. Erlhagen (2004). Coordinated transportation with minimal explicit communication between robots. *5th IFAC Symposium on Intelligent Autonomous Vehicles*.
- Bicho, E., L. Louro, N. Hipólito, S. Monteiro and W. Erlhagen (2003). Motion control of a mobile robot transporting a large size object in cooperation with a human: a nonlinear dynamical systems approach. *11th International Conference on Advanced Robotics (ICAR2003)*.
- Bicho, E., P. Mallet and G. Schöner (2000). Target representation on an autonomous vehicle with low level sensors. *The International Journal of Robotics and Research* **19**, 424–447.

- Chaimowicz, L., M. Campos and V. Kumar (2002). Dynamic role assignment for cooperative robots. *in Proc. of the 2002 IEEE Int. Conf. on Robotics and Automation* **1**, 293–298.
- Chaimowicz, L., T. Sugar, V. Kumar and V. Campos (2001). An architecture for tightly coupled multi-robot cooperation. *in Proc. IEEE Int. Conf. Robotics and Automation* **17**, 2292–2297.
- Crawford, J. (1991). Introduction to bifurcation theory. *Reviews of modern physics* **63**, 991–1037.
- Hashimoto, M., F. Oba and S. Zenitani (1993a). Coordinative object-transportation by multiple industrial mobile robots using coupler with mechanical compliance. *Proceedings of the IECON '93* **3**, 1577–1582.
- Hashimoto, M., F. Oba and S. Zenitani (1995). Object-transportation control by multiple wheeled vehicle-planar cartesian manipulator systems. *IEEE International Conference on Robotics and Automation* **3**, 2267–2272.
- Hashimoto, M., F. Oba, K. Nakahara and T. Eguchi (1993b). Dynamic control approach for motion coordination of multiple wheeled mobile robots transporting a single object. *in Proc. on Conf. on Intelligent Robots and Systems* **3**, 1944–1951.
- Hashimoto, M., F. Oba, K. Nakahara, K. Imamaki and T. Eguchi (1991). Trajectory generation and tracking control methods for a multiple transfer robots system. *Proc. on Conf. on Intelligent Robots and Systems* **2**, 799–804.
- Koga, M., K. Kosuge, K. Furuta and K. Nosaki (1992). Coordinated motion control of robot arms based on virtual internal model. *IEEE Trans. Robot Automation* **8**, 77–85.
- Kosuge, K., H. Takeda and Y. Hirata (2000). Decentralized motion control of two tracked mobile robots transporting a single object in coordination based on function allocation concept. *Aro Workshop 2000 on Intelligent Systems*.
- Kosuge, K., T. Oosumi, M. Satou, K. Chiba and K. Takeo (1998). Transportation of a single object by two decentralized-controlled nonholonomic mobile robots. *in Proc. of the 1998 IEEE Int. Conf. on Robotics and Automation* **4**, 2989–2994.

- Kume, Y., Y. Hirata, Z. Wang and K. Kosuge (2002). Decentralized control of multiple mobile manipulators handling a single object in coordination. *in Proc. of the 2002 IEEE/RSJ Intl. Conf. on Intelligent Robots and Systems*.
- Lewis, A. and K. Tan (1997). High precision formation control of multiple robots using virtual structures. *Autonomous Robots* **4**, 387–403.
- Louro, Luís (2003). Transporte de um objecto por uma equipa de dois robôs autónomos. *Relatório de Estágio - Universidade do Minho - Escola de Engenharia*.
- Nguyen, V. (1986). Constructing force - closure grasps. *in Proc. 1986 IEEE Int. Conf. on Robotics and Automation* **3**, 1386–1373.
- Pereira, G., B. Pimentel, L. Chaimowicz and M. Campos (2002). Coordination of multiple mobile robots in an object carrying task using implicit communication. *in Proc. of the 2002 IEEE/RSJ Int. Conference on Robotics and Automation* **1**, 281–286.
- Perko, L. (1991). *Differential equations and dynamical systems*. Berlin: Springer Verlag.
- Pimentel, Bruno S., Guilherme A. S. Pereira and Mario M. F. M. Campos (2002). On the development of cooperative behavior-based mobile manipulators. In: *AAMAS*. pp. 234–239.
- Schöner, Gregor, Michael Dose and Christoph Engels (1995). Dynamics of behavior: Theory and applications for autonomous robot architectures. *Robotics and Autonomous Systems* **16**(2-4), 213–245.
- Soares, R., E. Bicho, T. Machado and W. Erhagen (2007). Object transportation by multiple mobile robots generated by attractor dynamics: theory and implementation. *IEEE/RSJ Internacional Conference on Intelligent Robots and Systems*.
- Soares, Rui (2007). Transporte de objectos por equipas de robôs móveis autónomos: estratégias de controlo distribuídas baseadas em sistemas dinâmicos não lineares. PhD thesis. Universidade do Minho.
- Velasquez, C., T. Takahashi and E. Nakano (2004). Graspless coordinated transportation over natural flat terrain. *in Proc. of the 2004 Conf. on Robotics, Automation and Mechatronics* **1**, 165–170.

- Zaerpoor, A., M. Ahmadabadi, M. Baruni and Z. Wang (2003). Extending constrain-and-move strategy to move objects on a desired path by a team of distributed robots. *in Proc. of ICAR 2003 The 11th Int. Conf. on Advanced Robotics*.
- Zaerpoor, A., M. Ahmadabadi, M. Baruni and Z. Wang (2005). Distributed object transportation on a desired path based on constrain and move strategy. *Robotics and Autonomous Systems* **50**, 115–128.